PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-031127
(43)Date of publication of application : 02.02.1999

(51)Int.Cl. G06F 15/00
G06F 15/00
G06F 13/00
// G09C 1/00
G09C 1/00

(21)Application number : 10-126576
(22)Date of filing : 01.04.1998

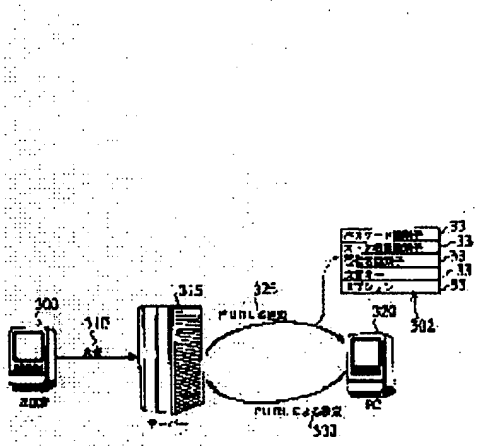
(71)Applicant : TUMBLEWEED SOFTWARE CORP
(72)Inventor : SMITH JEFFREY C
BANDINI JEAN-CHRISTOPHE

(30)Priority

Priority number : 97 829976 Priority date : 01.04.1997 Priority country : US
97 832784 04.04.1997 US

(54) DOCUMENT DELIVERY SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a method and a system which safely send a document to a communication network such as the internet.
SOLUTION: A document sending architecture dynamically generates a private uniform resource locator (URL) (PURL) 302 to distribute information. Each PURL 302 confirms a parameter of free selection which is peculiar to a receiver 320 of a document 310, a sending document 310 and a sending process on its own terms. The receiver 320 retrieves the document 310 with the PURL 302. A sender 300 commands a sending server 315 to retrieve the public key of the receiver 320. The server 315 dynamically collates an insurance source and retrieves the public key. The public key is sent from the server 315 to the sender 300. The sender 300 enciphers a secret key with the public key after enciphering the document 310 with the secret key. The document 310 and secret key which are enciphered are uploaded to the server 315 and sent to the receiver 320. The receiver 320 decodes the secret key with a decoding key that is related to the public key and decodes the document 310 with the secret key.



LEGAL STATUS

[Date of request for examination] 03.10.2000

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－31127

(43)公開日　平成11年(1999)２月２日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　15/00	３１０	Ｇ０６Ｆ　15/00	３１０Ａ
	３３０		３３０Ａ
13/00	３５１	13/00	３５１Ｅ
// Ｇ０９Ｃ　1/00	６３０	Ｇ０９Ｃ　1/00	６３０Ｆ
	６６０		６６０Ｅ

審査請求　未請求　請求項の数63　ＯＬ　外国語出願　（全103頁）

(21)出願番号　特願平10－126576

(22)出願日　平成10年(1998)４月１日

(31)優先権主張番号　０８／８２９９７６

(32)優先日　1997年４月１日

(33)優先権主張国　米国（ＵＳ）

(31)優先権主張番号　０８／８３２７８４

(32)優先日　1997年４月４日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　597150049
タンブルウィード　ソフトウェア　コーポレイション
アメリカ合衆国　カリフォルニア州　94063　レッドウッド　シティー　ブロードウェイ　2000

(72)発明者　ジェフリー　シー　スミス
アメリカ合衆国　カリフォルニア州　94025　メンロ　パーク　アルトシュール　アベニュー　1305

(74)代理人　弁理士　中村　稔　（外７名）

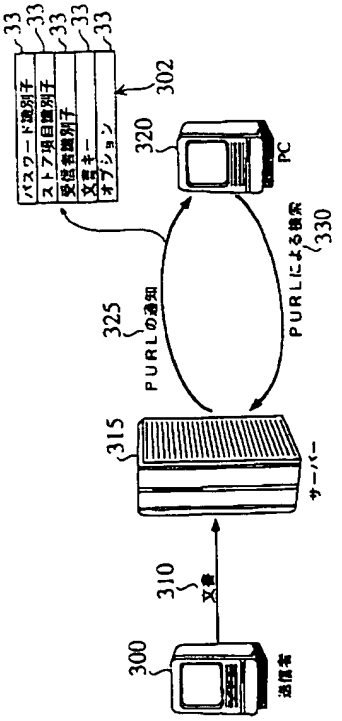
最終頁に続く

(54)【発明の名称】　ドキュメントデリバリシステム

(57)【要約】

【課題】　文書を安全に送付する方法とシステムを、インターネットのような通信網に提供する。

【解決手段】　文書送付アーキテクチャーが情報を配布するためのプライベートなＵＲＬ（ＰＵＲＬ）を動的に生成する。各ＰＵＲＬは、文書の受信者、送付文書、送付プロセスに固有な自由選択のパラメータを独自に確認する。受信者はＰＵＲＬで文書を検索する。送信者は送付サーバーに受信者の公開鍵の検索を命ずる。送付サーバーは保証元を動的に照会し、公開鍵を検索する。公開鍵が送付サーバーから送信者に送られる。送信者はシークレット鍵で文書を暗号化後、公開鍵でシークレット鍵を暗号化する。暗号化された文書とシークレット鍵を送付サーバーにアップロードし、受信者に送信する。受信者は公開鍵と関連する復号鍵でシークレット鍵を解読し、シークレット鍵で文書を解読する。



【特許請求の範囲】

【請求項１】　送信コンピューターと受信コンピューター間で電子文書を送付する装置において、前記送信コンピューターと前記受信コンピューター間に設置されたサーバーを有し、前記電子文書が前記コンピューターから前記サーバーへ送られ、前記サーバーがプライーベートなユニフォームリソースロケータ（「ＰＵＲＬ」）を動的に生成し前記電子文書を配布することを特徴とするサーバーを含む装置。

【請求項２】前記ＰＵＲＬが、前記電子文書の対象受信者と、前記電子文書の送付に特有な他の自由選択のパラメーターとを独自に確認することを特徴とする請求項１に記載の装置。

【請求項３】　前記電子文書の前記対象受信者が、前記ＰＵＲＬを使って前記電子文書を検索するのを特徴とする請求項２に記載の装置。

【請求項４】　前記サーバーが前記電子文書の検索時に、前記ＰＵＲＬに含まれる属性に基づいて前記検索の挙動をカスタマイズし、データベースの前記検索に関連するログ情報を自由選択でカスタマイズして、安全な文書送付と文書受信の追跡を可能にすることを特徴とする請求項３に記載の装置。

【請求項５】　前記のサーバーが、前記電子文書を送付するためのＨＴＴＰと、前記電子文書の前記サーバーへの到着を通知するためのＳＭＴＰ／電子メールとを使うことを特徴とする請求項１に記載の装置。

【請求項６】　前記ＵＲＬが、前記電子文書の対象受信者を独自に確認する一時的かつ動的に生成されたＵＲＬと、自由選択の前記電子文書自身と、前記電子文書に関連する自由選択の属性とを含むことを特徴とする請求項１に記載の装置。

【請求項７】　前期ＰＵＲＬが、送信される電子文書に汎用参照番号を付して、受信者が前記参照番号を経て前記電子文書にアクセスするのを可能にするのを特徴とする請求項１に記載の装置。

【請求項８】　前記受信者が前記参照番号を使用して前記文書にアクセスした時に、前記サーバーが、前記電子文書にアクセスする要求を傍受し、前記アクセスに関連して付加価値のあるサービスを提供することを特徴とする請求項７に記載の装置。

【請求項９】　前記ＰＵＲＬが、前記サーバーの文書のロックを解除するキーを更に含むことを特徴とする請求項１に記載の装置。

【請求項１０】　前記ＰＵＲＬが、前記電子文書の受信者を確認する独自の登録番号を更に含むことを特徴とする請求項１に記載の装置。

【請求項１１】　前記サーバーが、特定の個人が特定の文書にアクセスしたことに気付き、データベースにアクセスのあったことを記して文書追跡を提供することを特徴とする請求項１０に記載の装置。

【請求項１２】　送信者と少なくとも一人の受信者の間で電子文書を送付するための文書デリバリーシステムにおいて、前記電子文書を一時的に記憶する文書サーバーを有し、前記サーバーは、前記文書が送られる各対象受信者のためにプライベートかつ追跡可能なＵＲＬ（「ＰＵＲＬ」）を動的に作成することを特徴とする文書デリバリーシステムシステム。

【請求項１３】　前記サーバーが、デリバリーパラメーターないし前記ＰＵＲＬのトランザクション識別子を暗号化することを特徴とする請求項１２に記載のシステム。

【請求項１４】　前記ＰＵＲＬが、Ｅ－メールメッセージを有することを特徴とする請求項１２に記載のシステム。

【請求項１５】　前記受信者が、前記ＰＵＲＬを前記サーバーに示すことにより前記ＰＵＲＬを経て前記電子文書にアクセスすることを特徴とする請求項１２のシステム。

【請求項１６】　パスワードを示してからでないと前記ＰＵＲＬの参照符の付された電子文書にはアクセスできない旨を前記ＰＵＲＬが記すよう、前記サーバーが要求することを特徴とする請求項１５に記載のシステム。

【請求項１７】　前記サーバーが、前記ＰＵＲＬで前記電子文書にアクセスする特定の受信者を確認することを特徴とする請求項１５に記載のシステム。

【請求項１８】　前記サーバーが、特定の受信者が特定の文書にアクセスを試みた事実を記録することを特徴とする請求項１７に記載のシステム。

【請求項１９】　前記サーバーが、全ての電子文書が首尾よく送付された事実を記録することを特徴とする請求項１７に記載のシステム。

【請求項２０】　誰が前記電子文書にアクセスしたか、いつ文書にアクセスしたか、及び前記電子文書に首尾よくアクセスしたか否かのいずれをも述べた完全な記録を所有する前記サーバー上に維持させたないし関連させたデータベースを更に含むことを特徴とする請求項１２に記載のシステム。

【請求項２１】　前記サーバーの記録した情報が、電子文書の送信者に報告されることを特徴とする請求項２０に記載のシステム。

【請求項２２】　前記サーバーが、文書を検索する所定の受信者に関連した全ＩＰアドレス、前記同一ＰＵＲＬで所定の文書へ引き続いてなされた全アクセスのＩＰアドレス、ないし特定の受信者を目的とした特定の文書にアクセスしたＩＰアドレスを含むリストのいずれかを記録できることを特徴とする請求項１２に記載のシステム。

【請求項２３】　受信者が前記電子文書にアクセスし解読するための有効キーを示すまで前記電子文書が前記サーバー上で暗号化されたままであり、前記キーが前記Ｐ

ＵＲＬの部分中に暗号化されて示されることを特徴とする請求項１２に記載のシステム。

【請求項２４】　前記ＰＵＲＬが、キーが検索されなければならないことを記し、前記サーバーが受信者は前記電子文書を解読するための独自のパスワードを示すよう要求することを特徴とする請求項１２に記載のシステム。

【請求項２５】　前記ＰＵＲＬが、パスワードが所定の文書にアクセスするのに必要か否かを記すパスワード識別子を更に含むことを特徴とする請求項１２に記載の装置。

【請求項２６】　前記ＰＵＲＬが、所定の受信者がどの文書を得たいのかを独自に確認するストア項目識別子を更に含むことを特徴とする請求項１２に記載のシステム。

【請求項２７】　前記ＰＵＲＬが、所定の文書の対象受信者を独自に確認する受信者識別子を更に含むことを特徴とする請求項１２に記載のシステム。

【請求項２８】　前記ＰＵＲＬが、前記ＰＵＲＬ自体を有効にする文書キーを更に含むことを特徴とする請求項１２に記載のシステム。

【請求項２９】　前記文書キーが、所定の受信者に関連して無作為に生成された番号であるキーと所定の受信者が得たい文書を更に含み、前記キーが、所定の受信者確認番号が有効か否か、所定のストア確認番号が有効か否か、及び所定のストア確認番号を有する所定の受信者が文書へのアクセスするのを承認すべきか否かを確認するため使われることを特徴とする請求項２８に記載のシステム。

【請求項３０】　電子文書を送信者と少なくとも一人の受信者の間で送付する方法において、前記電子文書を一時的に保管するため文書サーバーを使い、前記サーバーが、前記文書が送られる各対象受信者のためにプライベートかつ追跡可能なＵＲＬ（「ＰＵＲＬ」）を動的に生成するステップと、前記ＰＵＲＬをそのコンポーネントの部分に解読するステップと、前記ＰＵＲＬの各コンポーネントの部分を有効にするステップを含むことを特徴とする方法。

【請求項３１】　キーを使用してＰＵＲＬを認証する段階を更に含むことを特徴とする請求項３０に記載の方法。

【請求項３２】　どのユーザーが文書にアクセスしているかを、所定の文書の対象受信者を独自に確認する受信者識別子を使って決定する段階を更に含むことを特徴とする請求項３１に記載の方法。

【請求項３３】　どの文書にユーザーがアクセスするかを、どの文書を所定の受信者が得たいかを独自に確認するストア項目識別子を使って決定する段階を更に含むことを特徴とする請求項３２に記載の方法。

【請求項３４】　前記文書が、送付される前に追加入力を要求するか否かを決定する段階を更に含むことを特徴とする請求項３３に記載の方法。

【請求項３５】　前記文書を前記受信者に送付する段階を更に含むことを特徴とする請求項３５に記載の方法。

【請求項３６】　アクセスの時間、送信の成功、及び受信者のＩＰのいずれをも含むデリバリートランザクションの属性全てを記録する段階を更に含むことを特徴とする請求項３１に記載の方法。

【請求項３７】　広範なネットワークで送信者から安全な送付文書をするための方法において、送信者がシークレットキーを使って文書を暗号化するステップと、送信者がデリバリーサーバーに交信し対象受信者に関連するパブリックキーを照会するステップと、デリバリーサーバーがリアルタイムでパブリックキーを動的に検索するステップと、デリバリーサーバーが送信者にパブリックキーを送信するステップと、送信者がパブリックキーでシークレットキーを暗号化するステップと、送信者が暗号化された文書と暗号化されたシークレットキーを受信者への送信用デリバリーサーバーに送信するステップとを含むことを特徴とする方法。

【請求項３８】　受信者がプライベートキーを使用してシークレットキーを解読する段階を更に含むことを特徴とする請求項３７に記載の方法。

【請求項３９】　受信者がシークレットキーを使って文書を解読する段階を更に含むことを特徴とする請求項３８に記載の方法。

【請求項４０】　送信者が、デリバリーサーバーからパブリックキーを受け取る前に文書を暗号化することを特徴とする請求項３７に記載の方法。

【請求項４１】　送信者が、デリバリーサーバーからパブリックキーを受信するのに続いて文書を暗号化することを特徴とする請求項３７に記載の方法。

【請求項４２】　文書が、データの同一限界内のデータ集合、データストリーム、ビデオ、音声データ、アニメーション、フォーマット化された文書ないしデータベースのいずれかであることを特徴とする請求項３７に記載の方法。

【請求項４３】　送信者が、対象受信者のアドレスと文書送付の指示をデリバリーサーバーに送るステップを更に含むことを特徴とする請求項３７に記載の方法。

【請求項４４】　広範なネットワークがインターネットであることを特徴とする請求項３７に記載の方法。

【請求項４５】　受信者が、デスクトップコンピューター、プリンター、ファックス機、パーソナルデジタルアシスタントないしネットワークコンピューター装置のいずれかであることを特徴とする請求項３７に記載の方法。

【請求項４６】　送信者が、デスクトップコンピューター、インターネットブラウザ装置、インターネット電話装置或いはネットワークコンピューター装置のいずれか

であることを特徴とする請求項３７に記載の方法。

【請求項４７】　データベースサーバーが、証明元、インターネットサーバー、パーソナルデジタルアシスタント、ないし対象受信者のデスクトップコンピューターのいずれか、ないし対象受信者のデスクトップコンピューターに接続されたイントラネットサーバーからパブリックキーを動的に検索することを特徴とする請求項３７に記載の方法。

【請求項４８】　広範なネットワークで送信者からの安全な文書送付をするための方法において、送信者がデリバリーサーバーに交信し文書の対象受信者に関連したパブリックキーを照会するステップと、デリバリーサーバーがリアルタイムでパブリックキーを動的に検索するステップと、デリバリーサーバーが送信者にパブリックキーを送信するステップと、送信者がパブリックキーで文書を暗号化するステップと、送信者が受信者への送信のためにデリバリーサーバーへ暗号化された文書を送信するステップとを含むことを特徴とする方法。

【請求項４９】　受信者が、プライベートキーを使用して文書を解読するステップを更に含むことを特徴とする請求項４８に記載の方法。

【請求項５０】　受信者が、デスクトップコンピューター、プリンター、ファックス機、パーソナルデジタルアシスタントないしネットワークコンピューター装置のいずれかであることを特徴とする請求項４８に記載の方法。

【請求項５１】　送信者が、デスクトップコンピューター、インターネットブラウザ装置、インターネット電話装置ないしネットワークコンピューター装置のいずれかであることを特徴とする請求項４８に記載の方法

【請求項５２】　データベースサーバーが、証明元、インターネットサーバー、パーソナルデジタルアシスタント、対象受信者のデスクトップコンピューターのいずれか、ないし対象受信者のデスクトップコンピューターに接続されたイントラネットサーバーからパブリックキーを動的に検索することを特徴とする請求項４８に記載の方法。

【請求項５３】　広範なネットワークで送信者からの安全な文書送付するための方法において、送信者がデリバリーサーバーに交信し対象受信者に関連したパブリックキーを照会するステップと、デリバリーサーバーがリアルタイムでパブリックキーを動的に検索するステップと、送信者が文書をデリバリーサーバーに送信するステップと、デリバリーサーバーがシークレットキーで文書を暗号化しかつパブリックキーでシークレットキーを暗号化するステップと、デリバリーサーバーが暗号化されたシークレットキーと暗号化された文書を対象受信者に送信するステップを含むことを特徴とする方法。

【請求項５４】　受信者が、プライベートキーを使用してシークレットキーを解読する段階を更に含むことを特徴とする請求項５３に記載の方法。

【請求項５５】　受信者が、シークレットキーを使用して文書を解読する段階を更に含むことを特徴とする請求項５４に記載の方法。

【請求項５６】　受信者が、デスクトップコンピューター、プリンター、ファックス機、パーソナルデジタルアシスタントないしネットワークコンピューター装置のいずれかであることを特徴とする請求項５３に記載の方法。

【請求項５７】　送信者が、デスクトップコンピューター、ネットワークコンピューター装置、インターネットブラウザ装置、インターネット電話装置ないしファックス機のいずれかであることを特徴とする請求項５３に記載の方法。

【請求項５８】　データベースサーバーが、証明元、インターネットサーバー、パーソナルデジタルアシスタント、指定された受信者のデスクトップコンピューターのいずれかから、ないし対象受信者のデスクトップコンピューターに接続されたイントラネットサーバーからパブリックキーを動的に検索することを特徴とする請求項５３に記載の方法。

【請求項５９】　前記受信者が前記検索時にパブリックキーを持っていない場合、前記デリバリーサーバーでパブリックキーを動的に生成する段階を更に含むことを特徴とする請求項５３に記載の方法。

【請求項６０】　前記動的生成ステップが、メッセージを前記受信者に送るステップを更に含み、当該メッセージの読み込みに当って前記受信者のシステム上でプライベート／パブリックキー一対を作成するモジュールを検索することを特徴とする請求項５９に記載の方法

【請求項６１】　前記動的生成ステップが、前記パブリックキーを前記受信者のシステムから前記デリバリーサーバーへ送るステップを更に含むことを特徴とする請求項６０に記載の方法。

【請求項６２】　広範なネットワークで送信者からの安全な文書送付をするためのシステムにおいて、対象受信者に関連したパブリックキーを送信者の指示で照会しかつリアルタイムでパブリックキーを動的に検索してパブリックキーを送信者に逆送信するデリバリーサーバーと、シークレットキーを使用して文書を暗号化しかつパブリックキーでシークレットキーを暗号化しかつ暗号化された文書と暗号化されたシークレットキーを対象受信者への送信のためデリバリーサーバーへ送信する送信者とを含むことを特徴とするシステム。

【請求項６３】　プライベートキーを使う受信者がシークレットキーを解読するための手段と、シークレットキーを使って暗号化された文書を解読するための手段とを更に含むことを特徴とする請求項６２に記載のシステム。

【発明の詳細な説明】

【０００１】
【産業上の利用分野】本発明は、コンピューターネットワーク分野に関する。より詳しくは、インターネットでユーザーに電子文書を送付する技術に関する。本発明は更に、インターネットの様に広範なネットワークでの安全な文書送信を提供するための方法とシステムに関する。
【０００２】
【技術的背景】例えばインターネットや他のオンライン発信源から提供されるコンピューター化された情報源の発達は、電子的に入手可能な情報を増大させた。現在では、インターネット加入ユーザーは、手動操作でインターネット中を駆けめぐり、関心のあったりなかったりする個所を訪れる。このインターネットシステムに固有の問題は、入手可能な情報が「プル」型基盤を通して配布されることであり、その場合、情報を受信したいユーザーは手動で関心のある部分を探すかファインダーアプリケーションを使うかして、適切な情報を探しダウンロードせねばならない。情報や文書を発行・配布したいユーザーにしてみれば、配布したい情報を持っている個人にしろ大きな団体にしろ、現行の「プル」型システムは、「プッシュ」方式で、受信者個人或いは受信者のグループへ自由に送信・配布することを許さない。
【０００３】ファクシミリ技術は、単純な文書の配布に広く現在使われているが、低品質の印刷文書、高価でかさばる紙コピー（特に受信者が紙コピーを持つのを気にしない場合）、内容の欠落（例えば、テキストやグラフは編集も手を加えることもできない）、特に長い或いは複雑な文書を送信するのに時間が掛かる等多くの欠点がある。電子メール（Ｅ－メール）は、電子メッセージをコンピューターユーザーから他の人へ送る手段を提供する。Ｅ－メールには、利便性、フォーマット、後の検索のためのメッセージの保存の利点がある。それなりにＥ－メールは受け入れられ、基本的な通信として広く使われている。Ｅ－メールはＡＳＣＩＩベースのフォーマットが代表的であるが、長い或いはフォーマット化された文書の通信をひどく制限することが判明している。更にＥ－メールは、複雑な文書、例えば、レポート、記事、ページレイアウト格子を含む広告と芸術、ポストスクリプト書式の対象物、トラッキングとカーニングのある複数フォント、グラフ、組み込まれた表、スプレッドシート、その他の複雑な情報の配布に当たって選ぶ手段ではない。Ｅ－メールシステムの中には、ＡＳＣＩＩベースのＥ－メールメッセージをダウンロードされる関連ファイルに追加する手段を提供するものがある。関連ファイルの追加が可能なシステムの殆どは、一人のユーザーから同僚ないし友人へ未保護ファイルの送信を可能とはするが、複数の受信者への制御された自動配布のためではなく、高度なアカウンティング、ビリング、ないしそうした特徴を備えたもの（例えば、領収通知書）を提供しない。Ｅ－メールゲートウエイでは、アタッチメントの応用性を制限されるし、機密保護と受信表示ないし受信通知の問題を未解決である。
【０００４】Ｃ．ボードインの米国特許番号５、４０６、５５７（１９９５年４月１１日）「企業間電子メールハブ」は、共有コアと複数の入出力モジュールから成るコンピューターハブを持つ企業間通信センターを開示している。入力モジュールは、第一エンドユーザーに接続され、第一エンドユーザーが送信したメッセージを汎用フォーマットに変換する。ハブコアはメッセージを待ち行列に入れ、行先ユーザーのフォーマットに変換すべく出力モジュールへ送る。開示されたハブは、単純なＥ－メールメッセージの中継技術を開示している一方で、Ｅ－メールメッセージフォーマットを変換する様設計されているため、本来のテキストベースのファイルの完全性を失っている。開示された先行技術システムと方法論はこのように、文書を送付する幾つかの方法を提供しているが、本来の電子ファイルの完全性を維持しつつプッシュ型で作動する経済的かつ迅速な文書送付システムを提供できていない。こうした電子文書送付システムが開発されれば、大きな技術進歩となろう。加えて、制御されかつ経済的で説明可能なやり方で多くの受信者に電子ポータブル高内容品質の文書を配布できる能力は、更なる技術進歩を構成するであろう。
【０００５】インターネットは、通信にますます使用されている。送信者は今やインターネットで、プラットフォーム、基本ソフトウェア、Ｅ－メールシステムとは無関係に、特定の受信者に文書を送ることができる。送信者のコンピューターのインターネットへの接続は、直接でもイントラネットのサーバー経由でもよい。こうした通信は、たとえ受信側がコンピューターでなく、インターネットに接続されたファックス機械ないしプリンターであっても可能である。インターネット通信の増加により、インターネットで送信される情報の保護を保証する機密保護システムの発達を必要とするに至った。暗号化は、情報への一方的なアクセスを防ぐため情報にスクランブルをかけるのに使用される基本技術である。よく知られている一つの暗号化方法はシークレットキー暗号化であり、時にプライベートキー暗号化ないし対称キー暗号方式とよばれる。シークレットキー暗号化で使われる技術は、一方的なアクセスを防ぐために独自キーを使って情報にスクランブルをかける。この独自キーは、情報を復元する際必要となる。図１は、先行技術に依るシークレットキー暗号化を示す図である。
【０００６】文書１０１０は、シークレットキー１０１４使って１０１２にスクランブルされる。シークレットキーは、スキームの公認ユーザーにのみ入手可能な暗号化スキームである。暗号化ソフトウエアは、ユーザーのコンピューター上或いは離れた位置にあってもよい。文書はこうして、本来の場所、ないし別のコンピューター

への送信時に例えばイントラネットサーバーで暗号化されてもよい。結果的に暗号化された文書１０１６が、受信者に送信される。シークレットキー１０１４を使って１０１８のスクランブルを解除し、本来の文書１０１０を再生する。暗号化された文書に、シークレットキーなしではアクセスできない。解読ソフトウエアは、受信者のコンピューター、離れた位置のどちらにあってもよい。

【０００７】シークレットキー暗号化に関する潜在的問題の一つは、シークレットキーの安全な配布である。シークレットキーが安全でないチャンネルで送信される場合、機密保護の完全性が危険に晒される。多くの実際的なアプリケーションの場合、電話ないしファックスはシークレットキー送付に十分な機密保護を提供するが、文書は、カリフォルニア州レッドウッド市タンブルウイードのソフトウエアコーポレイションから入手可能なポスタといったメールスキームを使いインターネットで送付することができる。しかし事例の中には、キー配布をより安全かつより便利に行う方法をユーザーが要求している。別の暗号化方法は、パブリックキー暗号化方法である。パブリックキー暗号化方法では、送信者と受信者がそれぞれパブリックキー及びプライベートキーと呼ばれる一対のキーを所有する。キーペアーの所有者はパブリックキーを発行し、プライベートキーを秘密にしておく。

【０００８】送信者は、対象受信者が発行したパブリックキーを使い、情報を暗号化する。情報の解読には、受信者のプライベートキーを使う。このように、パブリックキー暗号化方式を使えば、プライベートキーを配布する必要がない。図２は、先行技術によるパブリックキー暗号化方式を示す図である。文書１０２０は、パブリックキー１０２４を使って１０２２にスクランブルされる。結果的に暗号化された文書１０２６は、次に受信者に送信される。プライベートキー１０３０を使ってスクランブルを解除し、本来の文書１０２０を再生する。パブリックキー暗号化に使われるキーは、非常に大きな数字である。パブリックキー暗号化方法は、暗号化と解読を実行するためのキー数字間に難解な数理的関係を利用する。結果的に、発行されたプライベートキーからプライベートキーを難なく引出すことはできない。

【０００９】文書が送信中に変えられていないこと、ないし、文書の送受信者を確認することがしばしば役に立つ。シークレット・パブブリックキー技術は、こうした確認を可能とする。しかし、パブリックキー暗号化アルゴリズムは一般に難解で、時間が掛かり過ぎ実際の使用がしばしばできない。シークレットキー暗号化方法はずっと早いが、安全にキーを送信することに関し欠点がある。パブリックキー／プライベートキー暗号化システムは、ガネサン・ヤクシャの米国特許番号５、５３５、２７６（１９９６年７月９日）「分離プライベートキー非対称暗号方式を使用による通信保護のための改良されたシステムと方法」に述べられている。しかし、ガネサンの暗号化スキームの場合、一時キーの生成に複雑な方法を使っており、複数の異なるユーザーに手動でパブリックキーを要求させる。

【００１０】鳥居の米国特許番号５、３１３、５２１（１９９１年５月１７日）「ＬＡＮにおけるファイル転送のためのキー配布プロトコル」は、サーバーに対し端末を認証するためにキー配布センターを使う。パストルの米国特許番号４、８５３、９６１（１９８９年８月１日）「信頼性のある文書認証システム」では、解読キーを含む文書認証システムが述べられている。コウドリー他の米国特許番号５、５０９、０７４（１９９６年４月１６日）「電子発行された資料を暗号作成プロトコルを使用して保護する方法」が教示する文書保護システムは、各文書要求を認証するためサーバー同志間の機密保護アクセス操作を含んでいる。しかし、これら先行技術のスキームは全て、証明を認証するのにユーザーが介在せねばならない。

【００１１】別の暗号化方法である電子封筒は、シークレットキーとパブリックキー暗号化の欠点がない。電子封筒を使用し、送信者はシークレットキーで文書を暗号化する。その後シークレットキーをパブリックキーで暗号化する。文書の受信者は、自分のプライベートキーでシークレットキーを解読し、シークレットキーで文書を解読する。現在、パブリックキーの発行のためにレジストリを利用できる。こうしたレジストリは、特定のパブリックキーが特別の存在に属していることを証明できる。例えば、証明元は、それらの特定のパブリックキーに存在を接続するのに使われるデジタル証明を発行・維持する。送信者はレジストリを照会し、要求されたパブリックキー情報を受け取らねばならない。この時間の掛かるプロセスは効率が悪く、特に送信者が多数の文書を異なる受信者に送信せねばならぬ時が特に効率が悪い。

【００１２】暗号化を目的として広範なネットワークでパブリックキーを自動的かつ動的に検索するためのシステムと方法を供給することは有利である。そうしたシステムと方法でサーバーを使って証明を検索し、ユーザーの介在を必要としなければ更に有利である。このシステムと方法で、サーバーが使用者にパブリックキーを戻すまで文書をサーバーに送信しないなら更に一つの利点となる。

【００１３】

【発明の概要】電子文書送付システムとその使用の方法を提供する。文書送付アーキテクチャが、情報を配布するためのプライベートなユニフォームリソースロケータ（ＵＲＬ）を動的に生成する。各々のプライベートなＵＲＬ（“ＰＵＲＬ”）は、独自に、文書の対象受信者、送付される文書或いは文書セット、（随意選択で）送付プロセスに特有な他のパラメーターを定める。文書の対

象受信者はＰＵＲＬを使って文書を検索する。文書検索の際、サーバーは、ＰＵＲＬに含まれる属性に基いて検索の挙動をカスタマイズし、同様にデータベースにある検索に関連するログ情報もカスタマイズする。こうしたＰＵＲＬの構造と使用により、安全な文書送付と文書受信の追跡が可能となる。

【００１４】本発明は、例えばインターネットの様な広範なネットワークでの安全な文書送付のための方法とシステムを提供する。文書はデリバリーサーバーを経て送信者から受信者に送付される。本発明の好適実施例では、送信者がデリバリーサーバーに、対象受信者のパブリックキー（証明）を検索するよう命じる。デリバリーサーバーは保証元を動的に問い合わせ、パブリックキーを検索する。パブリックキーがデリバリーサーバーから送信者に送信される。送信者は、シークレットキーを使い文書を暗号化する。その後シークレットキーをパブリックキーで暗号化する。暗号化された文書と暗号化されたシークレットキーの両方をデリバリーサーバーにアップロードし、対象受信者に送信する。次に、対象受信者はパブリックキーと関連したプライベートキーで、シークレットキーを解読し、シークレットキーで文書を解読する。

【００１５】本発明の代替の同程度好適実施例において、送信者はパブリックキーを使い、文書を暗号化する。その後、暗号化された文書は対象受信者に送信され、パブリックキーに関連するプライベートキーで解読される。更に別の実施例では、サーバーが暗号化のため文書をデリバリーサーバーに送信する。デリバリーサーバーが、リアルタイムで証明元に問い合わせ、パブリックキーを検索する。デリバリーサーバーは、シークレットキーを使って文書を暗号化後、パブリックキーでシークレットキーを暗号化する。その後デリバリーサーバーは、暗号化された文書と暗号化されたシークレットキーを対象受信者に送信する。デリバリーサーバーの照会が失敗した場合（所定のユーザーの証明が入手できない）、デリバリーサーバーは対象受信者のために新しいパブリックキーを動的に生成する。その後、この新たな証明が文書暗号化に使われる。

【００１６】

【実施例】バイナリーファイルデリバリーシステム１０は、会社、出版社、個人が文書を電子的に配布できるようにする。重要なのは、バイナリーファイルデリバリーシステム１０は、既存のウエブを基にした文書発行技術と異なり、指示どうりかつ安全な文書配布を可能にする。現在のウエブは、文書の顧客がサーバーから文書を見つけ検索せねばならぬプル発行型の環境として特徴付けられる。これとは対照的に、プッシュ発行型は、文書の作成者が顧客宛に文書を送付するのを可能とする。ファクシミリ（ファックス）、郵便サービス、電子メール（Ｅ－メール）は、全てプッシュ発行の例である。図３は、一台のバイナリーファイルサーバー１２を使用するバイナリーファイルデリバリーシステム１０を示すブロック図である。バイナリーファイルデリバリーシステム１０は、ユーザーが文書をプッシュするのを可能にし、文書作成者が文書の行先を指示するのを可能とする。バイナリーファイルデリバリーシステム１０がプッシュ発行を達成する一つの方法は、ネットワークで情報をプルするために通常実行されるＨＴＴＰを、ＳＭＴＰ（テキストのみをサポート）に結び合せる方法である。バイナリーファイルデリバリーサーバー１０は更に、指示された文書送付の様々なアプリケーションを容易にするため、サービスのホスト役を果たす。或るレベルでバイナリーファイルデリバリーシステム１０は、新世代のファクシミリ技術として特徴付けが可能で、電話回線の代わってネットワークを利用し更に既存のファックス様式より非常に優れた新しい文書表示のためのサポートを導入する。別のレベルで、バイナリーファイルデリバリーシステム１０は、膨大な量の文書と取引を支援できる汎用文書デリバリーサーバーである。あらゆるケースでバイナリーファイルデリバリーサーバー１０は、文書送付に関する完全で力強い解決策を提供する。

【００１７】バイナリーファイルデリバリーサーバー１０は、或るエンドポイントから一つ以上のエンドポイントへバイナリーファイルのセットを送信するのに使われる。エンドポイントは、インターネットアクセスする受信者２２が代表的であるが、ファクシミリ機械１７２又はプリンター１７８といった別のものでもよい（図１６、１７）。バイナリーファイルは、信頼性があり、説明可能で、かつ追跡可能な方法で送付される。バイナリーファイルデリバリーシステム１０が指示されたファイルに提供する機密保護のレベルは、Ｅ－メール同等からファクシミリないし一般の郵便よりは優れたレベルまで様々である。本システムは、ビリングアカウントの貸方・借方を始めとするユーザーアカウント管理を提供する。システムは複数のバイナリーファイルデリバリーサーバー１２の間で協力動作でき、サーバーは他の許可により制御されてもされなくてもよい。図４は、インターネットと通信する二台のバイナリーファイルサーバー１２ａ、１２ｎを使うバイナリーファイルデリバリーシステムを示す。

【００１８】バイナリーファイルデリバリーサーバー１２は三種の基本モードで作動し、これらモードは、送信者１６がアカウント１３２自体を設定しビリングに従属するパブリックモード、送信者１６が管理者に制御されビリングが徴収問題というよりもより内部アカウント問題であるプライベートモード、受信者２２は多いが送信者１６の少ない発行モードの三種である。バイナリーファイルデリバリーサーバー１２は、独立した機能コンポーネントから成り、必ずしもプロセスないし共有ライブラリではない。図６に概略で示されているバイナリーフ

ファイルサーバー１２は、ストア４２と呼ばれるインテリジェント記憶コンパートメントを含み、本装置はストアクライアント４４と呼ばれる一連のクライアント４４ａ－４４ｎで増大され、ストアクライアント４４はストア方法を使いストアイベントに従うが、他のクライアント４４と対話したり識別しあうことがない。アカウントマネージャ４６コンポーネントは、送信者１６に関する情報を保持する共有サービスである。受信アプリケーションの場合の、受信者２２に関する情報が更に設計上組み込まれている（Ｅ－メール通知とは対照的である）。

【００１９】クライアント／サーバーの汎用アーキテクチャは、よりパイプライン化された構造よりも、より優れた拡張性がある。しかも、本アーキテクチャによりストアクライアント４４は互いから切り離されるので、幾つかのタスクが相互作用し他がより背後志向であるような状況で有効である。

［ストア］　ストア４２は、一連のストア項目４８を含む。図５に示す様に、ストア項目４８は、バイナリーファイルトリー３４と記述子３６を含み、記述子はストア定義された属性とクライアント定義された属性の一連のセットである。バイナリーファイルトリー３４は、ストア定義された属性の部分と見なすことができる。ファイル記憶システムは次に述べる機能性を提供する。

【００２０】１）ストア項目４８の永久記憶（例えば、ストア項目４８に含まれるバイナリーファイルツリー３４がディスクに書き込まれる。）

２）ストア定義された属性とクライアント定義された属性からなる記述子３６へのクライアント読込／書込アクセス（例えば、クライアント４４は、ストア項目４８の満了日を書き込むことができる。）

３）ストアイベント６７のクライアント通知（例えば、クライアント４４は、新しいストア項目４８の作成イベント６８の通知を受けられる。）

４）ストア定義された属性による内部管理（例えば、ストア項目満了日は、イベントを生成する。）

ストア４２は、ストア項目４８へのアクセスを提供し、ストアイベント６７を作成し、ストア項目４８はストア定義された属性、例えばＩＤ、作成データ、ファイルカウント、ファイルネーム、ファイルデータを所有し、ストアイベント６７は、クライアント４４に従うはずである。ストアイベント６７は、ストア項目４８の作成６８、削除６９、ないし変更修正７０を含んでもよい。イベント６７はアーキテクチャにおける決定的役割を果たすが、何故なら、クライアント４４が自分の仕事を他者に関する非常に限られた情報とどのように同期化させるかを定義しているからである。

【００２１】［ストアクライアント］　ストアクライアント４４は非常に多様で、特定のクライアントについては更に詳述する。本構成の場合、ストアクライアント４４はある種のコンポーネントであって、ストア項目４８での役に立つタスクを実行するために、ストア方法の幾つかを使用するかないしストアイベント６７の幾つかに従う。

［アカウントマネージャー］　アカウントマネージャー４６は、ユーザーとビリングアカウントへの読み込み／書き込みアクセスを提供し、クライアント４４ないしシステム１０の他のコンポーネントが使用する。ストア４２はアカウントを使いも識別もしない。

［他のコンポーネント］　ストアクライアント４４とストア４２自身が使う他のコンポーネントは、本システムのアーキテクチャ内で実行される。例えば、相互サーバー通信、ログ管理、及び他の管理サービスでありこれらについて以下に述べる。

【００２２】図７は、バイナリーファイルサーバー４２の一つの実施例のアーキテクチャの例で、サーバー機能の実行に使われるクライアント４４モジュール（５２－６６）を含んでいる。インターネット送信５２で、ストア項目４８を作成し、属性を書き込む。インターネット受信５４で、既存のストア項目４８を開き、それら属性を変更できる。ファックスゲートウエイ５６で、ストア４２が生成した作成イベント６８に従い、関連したストア項目４８を処理し、次にストア４２からそれらを削除する。フォワーダー５８は、ストア４２が生成した作成イベント６８に従い、次に新たなストア項目４８の属性を調査し、フォワーディングが必要か否か決定する。アーカイバー６０は削除イベントに従い、削除が行われる前に第二プライベート記憶にストア項目４８をコピーする。フォーマットトランスレーター６２は作成に従い、属性を調査し、トランスレーションが必要ならば、読込、処理、ストア項目４８のファイルに書き戻しを行う。ウエブパブリシャー６４は作成イベント６８に従い、ストア項目属性がウエブパブリシングを記したかをチェックし、もし記していれば、必要と見なし属性を読み込む。ピックアップ通知６６は作成イベント６８に従い、受信者２２に通知する。

【００２３】［インターネットベースのユーザーのための機密保護問題］　バイナリーファイルデリバリーシステム１０が専用の機密保護解決策を支援する柔軟性を提供する一方、現在の産業ベースの機密保護解決策を難なく支援しており、それらとして以下のものを挙げることができる。

ａ）安全なサーバー間接続とサーバー認証（ＳＳＬ２．０で利用でき、サーバー（ＨＴＴＰ）に組み込まれている）

ｂ）サーバー間の機密保護（ＳＳＬＸ上で）

ｃ）支援エンドポイントプライベートキー（ユーザーは、自分のチャンネルを使ってプライベートキーを交換せねばならない）

ｄ）暗号化ＡＰＩないし標準ユーザーパブリックキーを使いエンドポイントパブリックキーを支援する。本シス

テムは更に、ユーザーがＢＦＤ使用のみに限ってパブリックキーを生成してそのキーでユーザーアカウント情報を更新するのを助けることもでき、送信者はパブリックキーを得るために受信者と直接通信せずに済む。

【００２４】ｅ）ＳＳＬとＭＳ　ＰＣＴを有するサーバーによるクライアント認証（エンドユーザーは自身の証明を得られ、サーバーにより認証してもらえる）

バイナリーファイルデリバリーサーバー１２の重要な面は、複数の要求を同時に処理し、殆どの要求に対する応答時間を最小にすることである。それ故、同期化の問題は正確さとシステム性能の両者にとり重要である。性能の向上には、同期化させたデータアクセスを最小にし、可能な時にはいつでも非同期処理に先送りし、多重タスキングとプラットフォームのためのプロセス内通信（ＩＰＣ）を用いる。サーバー１２の一実施例では、一プロセス内で低オーバーヘッド多重タスキングを提供するスレッディングを大きな拠り所とし、又多重プロセッサー性能を利用可能な時に導入する。本実施例のＩＰＣでは、メールスロットないし遠隔手続き呼び出し（ＲＰＣ）に加えて、ネームドパイプを使う。

【００２５】図７は、バイナリーファイルデリバリーサーバー１２のアーキテクチャ内の特定のコンポーネントのブロック図を示す。ユーザーセッション７２が取り扱うセッションは、送信セッション、受信セッション（ユーザーがＢＦＤデスクトップアプリケーション１９２、１９８を使う時に実行される）、ＨＴＭＬ受信セッション（ユーザーがＢＦＤデスクトップ１６４使う時とは対照的に、ＨＴＭＬブラウザーを通して実行される（ＢＦＤデスクトップセッションはＨＴＭＬを通して実行されてもよいことに留意））、保守セッション（アカウントセットアップと保守セッションを実行する（例えば、通知ダウンロード、アカウントセッティング変更修正、パブリックサーバーのエンドユーザーと対照的な管理者によるコンソールサービスと混同しない様に））、ＨＴＭＬ保守セッション（ＨＴＭＬブラウザを通してアカウントセットアップと保守を実行する）である。

【００２６】デリバリーコンポーネント７４は、通知やフォワーディングを含んだ送付という背後作業を実行する。コンソール７６は管理セッションを実行し、このセッションは専用のユーザーインターフェースの代わりにＨＴＭＬインターフェースを通してなされる。コンソール７６は、アカウント、ロギング、パフォーマンス、パラメーターセッティングを含む全てのサーバー属性をブラウズし変更するためのユーザーインターフェースを提供する。

［共有コンポーネント］　共有コンポーネントは、ストア４２、ストアクライアント４４のいずれで使用されてもよく、ないしそれ自体で作動してもよい。共有コンポーネントがストアイベント６７に従わない間は、必要に応じて効率、例えばコネクター受信のために、ストア方法を使用できる。共有コンポーネントには以下が含まれる。

【００２７】１）アカウントマネージャー。これは、全てのローカルアカウント情報を維持し、ビリングアカウント及び遠隔アカウント情報を始めとするローカルアカウントへの独自のアクセスインターフェースを提供する。

２）サーバー間の全通信を取り扱うサーバーコネクター８０。

３）出されたメールの送受信を取り扱うメールゲートウエイ８４。

４）ロガー８６。型毎に分類された異なるログへの読込／書込アクセスを管理する。最も重要なログは、ストア項目４８で何が起きたかを追跡する送信／受信トランザクションログである。

５）オペレーティングシステムアクセサー８２。これは、オペレーティングシステムが行うファイル入出力（Ｉ／Ｏ）、プロセス管理（同期化、ロッキング、スレッド、プロセス）、ＩＰＣ（ＲＰＣ、共有記憶、共有待ち行列、パイプ）、ネットワークアクセス（ＴＣＰ／ＩＰソケット、ＨＴＴＰサーバーインターフェーシング、ＰＯＰ／ＳＭＴＰインターフェーシング）用にプラットフォーム独立型インターフェースを提供する。特定の部分が必要に応じて実行される。

【００２８】［サーバーアプリケーション］　サーバーアプリケーション８８は、全てのバイナリーファイルデリバリーサーバー１２を構成パラメーターに従って立ち上げ又停止するのに使われる。更に、アカウントマネージャー（４６ないし７８）ないしロガー８６がカバーしないサーバーの管理面の役割、例えばパフォーマンスプロファイリング、使用法情報、サーバーパラメーター／構成も果たす。図１０は、ストア４２のアーキテクチャを示すブロック図である。ストアマネージャー９２は、包括的状態を維持し、ストア４２へのアクセスを同期化し、ハウスキーピング機能を提供するのに使われる。ストア項目マネージャー９４は、ストア項目４８の状態、ロック、キャッシュ機構を維持するために使われる。ストアイベントマネージャー９６は、リスナーリスト及びイベントフィルターを維持し、同様にイベントフィルターとイベント優先順位に従ってイベントをディスパッチするのにも使わる。

【００２９】図１１は、インターネットクライアントをセッション、トランザクション、トランスポートを含む三層にユーザーセッションがどう組織化するかを示す。セッションマネージャー１０２は、現在作動中の全セッション状態を維持し、セッションに関連したハウスキーピングを実行する。１０２は、ストア４２とアカウントマネージャー４６の使いトランザクションマネージャー１０８から届くトランザクションを処理する。トランザクションマネージャー１０８は、トランスポートマネー

ジャー１１４、１１８から未処理のデータを受信し、一つ以上のＢＦＤトランザクションインタープリター１１０ないしＨＴＭＬトランザクションインタープリター１１２を使ってバリデーションとプリプロセシングを実行する。トランザクションマネージャー１０８は、次にデータを適切なＢＦＤセッションマネージャー１０４ないしＨＴＭＬセッションマネージャー１０６に提出し、回答を待ち、次に回答を適切なトランスポートマネージャー１１４ないし１１８に戻す。

【００３０】図１２は、送信セッションがストア項目４８を一旦作成し、ないし別のサーバー１２ａ－ｎがストア項目４８を送っている時のデリバリーの非対話タスク１２０を示す。デリバリーマネージャー１２２は、関連するストアイベントに従い、フォワーディング決定を下し、作業をノーティファイアー６６とフォワーダー５８に調整する。サーバーディレクトリは、Ｅ－メールドメインとサーバードメインの間の関連付けを追跡し続ける。ノーティファイアー６６は、Ｅ－メール通知２０を受信者２２に渡すのに使われる。フォワーダー５８は、サーバーコネクター８０を使ってストア項目４８を他のサーバー１２ａ－ｎへ送るのに使われる。全ての通知が受信されるとは限らないので、Ｅ－メールスキャナーで「返却された」Ｅ－メールのサーバーメールアカウントをチェックし、それを失敗トランザクションとする。

【００３１】図１３は、アカウントマネージャーのアーキテクチャ１３０の詳細を示す。アカウントマネージャー７８は、ローカルサーバー１２のためにユーザーアカウント状態１３２を維持し、ローカルアカウント１３２のためにビリングアカウント状態１３４を保持し、ローカルアカウント１３２を照会し、リモートアカウント１３６のディレクトリーを維持するのに使われる。リモートアカウントディレクトリー１３６の第一目的は、Ｅ－メールアドレスをＢＦＤアカウントないし非ＢＦＤアカウントと関連付けることである。図１４は、ロガーアーキテクチャの詳細を示す。図１５は、サーバーコネクターアーキテクチャの詳細を示す。

［システムオペレーション］　次の例は、電子情報を送信者１６から受信者２２に配布するのにバイナリーファイルデリバリーシステム１０がどう使われているかを示す。仮の発行者、カリフォルニア州レッドウッド市のサムは、日本の東京にいる友人ロブに文書を送りたい。以下に進行するイベントで、制御された形でこれがどのように達せられるかを示す。

【００３２】［サムはカリフォルニア州サンタクララのローカルサーバーに接続する。］サムのＢＦＤデスクトップが、彼のユーザーアカウントがあるサンタクララのローカルサーバー１２ａに接続される。セッションマネージャー１０２は、ユーザー１６（サム）を有効にする様アカウントマネージャー７８に照会する。セッションマネージャー１０２は次にユーザー１６に対し送信セッション状態を作る。［サムの送信セッション］　サムのＢＦＤデスクトップがトランザクション詳細、例えばファイル番号、ファイルサイズ、対象受信者を送信する。セッションマネージャー１０２は、このデータをセッション状態に添える。次にセッションマネージャー１０２は、メモリーにあるストア項目４８記述子３６を作成し、ストア４２にディスクスペースを確保し、ストア項目ＩＤも又同様である。その後アップロードが始まる。セッションマネージャー１０２は、データを非同期Ｉ／Ｏで直接ファイルに受け渡しする。

【００３３】サムの全ファイルのアップロード１８が完了した時、セッションマネージャー１０２は、ストア項目記述子３６をディスクに対し非同期で更新し、次にストア項目４８をストア４２に非同期で挿入する。セッションマネージャー１０２は、サムのアップロードに認知の旨回答し、トランザクションに関する情報を提供する。こうして本セッションが終了する。

［サンタクララストアにて］　ストア項目４８の挿入は、非同期にストア４２によりロガー８６にログインされる。ストア４２は、次に、登録されたイベントハンドラーフィルターに反して、ストア項目記述子３６を作動する。整合毎に、ストア４２がイベントと受信者（ロブ）をイベント待ち行列に挿入する。こうしてスレッドは終わる。

【００３４】イベントディスパッチスレッドはイベントをプルし、そのイベントを非同期で受信者にディスパッチするが、ディスパッチの速度はシステムのチューニングパラメーターに依存する。

［サンタクララデリバリーが通知される。］　デリバリー７４は、関連するイベントを通知され、ストア項目４８のロックに応じるスレッドをストア４２との同期トランザクションを経て開始させる。一旦ロックが保全されると、スレッドがストア項目記述子３６を読み込み、記述子をどう扱うか決めるため、デリバリーマネージャー７４が分析する。受信者２２は別のＢＦＤ１２ｎの置かれている日本ドメインにいるとわかる。デリバリーマネージャー７４がサーバーディレクトリ１２４に照会することにより、これがわかる。マネージャーが次にストア項目４８を送ることを決める。フォワードマネージャー８０は、コネクター８０が東京に送るように非同期で依頼する。そうしてデリバリー内のスレッドが終了する。デリバリーがサーバープロトコルを識別してないことに留意されたい。

【００３５】サンタクララコネクター８０は、東京コネクター８０に送信を始める。デリバリー要求を扱うスレッドは、最終的にコネクター８０で始動する。コネクターはホストを知っており又ストア項目４８のロックを所有している。東京サーバー１２ｎとの接続を開始する。接続できない場合はしばらくの間休止する。最終的に接続が開き、コネクター８０がプロトコルインタープリタ

ーに入り、最終的にストア項目記述子３６と関連のバイナリーデータファイル３４とを転送する。その後接続を閉じ、東京サーバー１２ｎへの転送成功をロガー８６に記録する。送られた旨記録後、次にコネクター８０はストア４２中にあるストア項目４８のロックを開放する。ロックの解除時、ストア４２は、イベントフィルターリストに反してストア項目記述子３６を作動し、局地で扱われているイベントフィルターを見つける。首尾よく送られたストア項目４８は、１だけ減った参照番号を生じさせる。本例の場合、受信者は一人だけなので、カウント番号が０になることを意味する。それ故、ストア４２はストア項目４８を削除リストに移すことができる。ストア４２のハウスキービングスレッドが、ある時点でストア項目４８を取り除く。

【００３６】東京コネクターレシーバー８０のスレッドが、接続を扱うため開始される。ひとたびプロトコルインタープリターがフォワードと理解すると、ストア４２にストア項目ＩＤ３６と各自に付された記憶スペースとを要求する。実際のストア項目記述子３６とファイルは、ディスクがデータを受信している最中に、ディスクに書き込まれてしまう。ひとたび接続が完了すると、ストア項目４８は、東京のバイナリーファイルデリバリーサーバー１２ｎのストア４２に非同期で挿入される。

［東京デリバリー装置始動］　挿入時に東京のストア４２は、デリバリーのスレッドが取り扱おうとするイベントを生成する。新たな項目の挿入もロガー８６に記録する。デリバリー７４のマネージャー１０２は、これが送られたこととこのサーバー１２ｎから受信されるであろうことを了解する。

【００３７】サーバー１２ｎは、ロブのＥ－メールアドレスに関連したアカウントがあるか否かをアカウントマネージャー７８に照会する。ロブのＥ－メールに関連するアカウントがなければ、ストア項目ＩＤ３６を示すＵＲＬ付のＥ－メールがロブに送られる。又、ロブの通知を受けたことをコネクター８０がサンタクララサーバー１２ａに知らせるという非同期の要求を待ち行列に入れる。ロブがアカウントをここに持っていれば、デリバリーが、未決定のデリバリーを告げる様にアカウントマネージャー７８で非同期の更新要求を出す。この場合、シナリオはまだ続く。

［ロブが東京サーバーに接続し、新しい文書をチェックする。］　ロブが受信セッションを開く時、セッションマネージャー１０２はロブのアカウントの有効性を同期的にチェックし、その過程でセッション状態を更新し、アカウントに未決定受信の合図があることを記憶する。ロブのＢＦＤデスクトップは結果的に、文書が受信されるべきだと要求する。セッション状態は答を得て、はいと言う。

【００３８】ロブのデスクトップ１７０が受信を要求すると、セッションマネージャー１０２が、ストア４２に関連するストア項目４８のロックを同期的に要求する。ひとたび承諾されると、データの最初の部分を送信することで回答できる。ひとたび文書がダウンロードされると、受信の成功をロガー８６に非同期的に記録する。こうして、最終的送付をサンタクララサーバー１２ａに通知する様、コネクター８０で非同期の要求を出す。東京での受信セッションでは、セッションマネージャー１０２がロックを解除し、ストア４２に非同期削除要求を出す。ロブの受信セッションは、この時終了する。サンタクララのコネクター８０がプロトコルインタープリターを作動させると、通知をロガー８６に待ち行列に入れるべき旨インタープリタが告げる。

【００３９】［サムが状態をチェックする。］　サムは、保守セッションに続く受信セッションを行うために接続する。保守セッション７２は、送信された文書の状態をチェックする要求を受信する。保守セッション７２は、送信時にサムのデスクトップに伝えられたストア項目ＩＤ３６を使いロガー８６に同期的に照会を出す。照会はマッチング記録のリストを戻し、それらは処理されデスクトップに戻され、こうしてユーザーインターフェース１６を更新できる。

［ポータブル文書デリバリーシステム］　電子ポータブル文書はますます普及している。これらファイルは、本来の外観と感触を失うことなしに異なるプラットフォームに配布できる。アドベシステムのアクロバットＰＤＦ™とノーベルのＥｎｖｏｙ™のポータブル文書フォーマットが広く使われている。本発明の好適実施例では、ポータブル文書デリバリーシステム１６０が、ポータブル文書技術をインターネットに適用して、電子文書の配布に関し一般的な解決を果たした。ポータブル文書デリバリーシステム１６０は、ノーベルのＥＮＶＯＹ™とアドベシステムのＰＴＦ™両フォーマットを始めとするポータブル電子文書フォーマットとの完全な互換性を提供する。

【００４０】ポータブル文書デリバリーシステム１６０からポータブル文書を受信する受信者２２は、それら文書から情報を見て、検索し、印刷し、保管し、取り出すことができる。ポータブル文書デリバリーシステム１６０に関連しＥｎｖｏｙ™ないしＡｃｒｏｂａｔ™を使って配布された文書は、完全な視覚忠実性を保ち、最高レベルの品質と解像度で高解像度出力装置上に作成できる。ポータブル文書フォーマットは、文書内の情報の内容と色を保つことを可能とし、多くのフォーマットは、かさ張らない状態でファイルが保管できる一方、索引付け、検索、ハイパーテキストリンキングができる。図１４は、バイナリーファイルデリバリーサーバー１２を使用したポータブル文書デリバリーシステム１６０ａの機能ブロック図である。図１５は、インターネットで通信するバイナリーファイルデリバリーサーバー１２ａと１２ｎ二台を使ったポータブル文書デリバリーシステム１

６０ｂの機能ブロック図を示す。

【００４１】ウエブと電子メールの限界について追加的サービスに加えて述べると、ポータブル文書デリバリーシステム１６０は、既存の電子メール上、即ち、ｈｔｔｐサーバーソフトウエア上で作動するサーバーソフトウエアとデータベースシステムとを含む。このように、ポータブル文書デリバリーシステム１６０は、電子メール、ウエブ、データベースに関する業界標準の解決策を結合し、会社やユーザーが受信者に文書送付を伝えられる様にしている。次の開示は、一般的な文書デリバリー解決策に関する要件を、ポータブル文書デリバリーシステム１６０の特定コンポーネントと同様に詳しく述べている。ポータブル文書デリバリーシステム１６０は、三つの基本コンポーネントを結合し、一般的な文書デリバリーの解決策を提供する。

【００４２】１）ポータブル文書送信クライアント　ポータブル文書送信クライアント（ＰＤＳＣ）１９２は、全てのデスクトップアプリケーション１９０を直接ポータブル文書デリバリーシステム１６０に内蔵する。ＰＤＳＣ１９２は、本発明の全実施例に必要とは限らない。単にＢＦＤサーバー１２を直接導入したい発行者は、そうすればよい。ＰＤＳＣ１９２は、地点間のデリバリーに問題のある標準的な企業のコンピューターユーザーを対象としている。

２）バイナリーファイルサーバー　バイナリーファイルデリバリーサーバー１２は、インターネット標準上で作動し受信者に文書を送付する。ＢＦＤサーバー１２は、ポータブル文書送信クライアント（ＰＤＳＣ）１９２を通して明白に呼び出す、ないしサーバー構成ユーザーインターフェース１９８を使用して呼び出し直接カスタマイズすることができる。

【００４３】３）ポータブル文書受信クライアント　ポータブル文書受信クライアント（ＰＤＲＣ）１９４は、文書の受信者２２が文書を受け取り、見て、プリントするのに利用するソフトウエアコンポーネントである。ＰＤＲＣソフトウエア１９４を持たない受信者２２は、リンクが与えられてインターネットで直接ソフトウエアにアクセスする。殆どの場合、ＰＤＲＣ１９４は、ネットスケープナビゲーターTMプラグインないしマイクロソフトアクティブXTMコントロールないしジャバアプレットとして単純に機能し、こうして直接ＰＤＲＣ１９４は受信者の既存のブラウザに同化する。

【００４４】図１８は、ポータブル文書送信クライアントアプリケーションとポータブル文書受信クライアントアプリケーションが本発明でどう使われているか示す。図１９は、サーバー構成ユーザーインターフェースアプリケーションが本発明でどう使われているか示す。

［ポータブル文書デリバリーシステムの要件］　最も基本的なレベルとして、文書送付解決策は、文書の作成者により文書が受信者に宛てられるないし「ｐｕｓｈされる」のを可能とせねばならない。ポータブル文書デリバリーシステム１６０の設計ねらいは、様々なコンピューターシステム上で様々なオペレーティングシステム、Ｅーメールシステム、文書タイプを作動させる様々な受信者が、電子ポータブル文書を受け取り、読み、使用して利益を得られる様にすることである。ポータブル文書デリバリーシステム１６０が適合している様々な設計パラメーターカテゴリーとして、主要コンピューターシステム（例えば、パソコン、ワークステーション、サーバー）、主要オペレーティングシステム（例えば、マッキントッシュ、ウインドウズ３．１、ウインドウズ９５、ＮＴ、ユニックス、ＯＳ／２）、電子メールシステム（例えばマイクロソフト、ｃｃ：メール、グループワイズ、ノーツ、ユードラ）、文書タイプ（例えば、紙、ポストスクリプト、クアーク、ワードパーフェクト、エクセル）、及び使用者タイプ（例えば、ＭＩＳ、法律、財務、消費者／家庭、市場通信（ＭａｒＣｏｍ））が挙げられる。

【００４５】ポータブル文書デリバリー１６０の特有な面は、全てのコンピューターシステム、オペレーティングシステム、電子メールシステム、文書タイプに対して解決策がもたらす互換性のレベルである。本発明の一実施例の場合、文書の送信者１６と受信者２２の両者ともインターネットに接続している。本発明の好適実施例においては、ポータブル文書デリバリーシステム１６０は、インターネットでの送付問題の解決策のみならず、ファクシミリ機１７２とプリンター１７８とのバックワード互換性も、今後の配布プリントアーキテクチャとのフォワード互換性と同様に提供している。

［一般的なデリバリー］　送付問題解決策は、ユーザーが誰にでも文書を配布できるようにせねばならず、様々なコンピューティングープラットフォーム、ファクシミリ１７２との互換性、今後の配布された印刷アーキテクチャとの互換性への支援を必要とする。ポータブル文書デリバリーシステム１６０は、複雑なポストスクリプトファイルの変換と送付を支援できる。文書は、Ｅーメールアカウントを持ちインターネットにアクセスする受信者２２へなら誰へでも、受信者のプラットフォームないしＥーメールシステムを問わず、送付できる。

【００４６】［機密保護］文書送付の代表的アプリケーションは、文書全体の源から目的場所までの完全な機密保護を要求する。開放されかつ広範なネットワーク中を文書が流れ始めるにつれ、この要件がより浸透してゆく。ポータブル文書デリバリーシステム１６０は、機密保護に複数のレベルを使用する。ポータブル文書送信クライアント１９２は、サーバー１２に情報をアップロードするために安全なソケットを認証し作成する。こうして非ＢＦＤサーバーは文書を傍受できなくなる。加えて、ＰＤＳＣ１９２は、文書の対象受信者にのみ文書へアクセス可能とするのを保証すべく、送信者１６がプラ

イベートないしパブリック暗号化方法を使える様にする。暗号化が使われない場合でも、ポータブル文書デリバリーシステム１６０は、未許可のユーザーが文書にアクセスするのを防ぐ精巧なアルゴリズムを含む。

【００４７】［アカウントマネジメントサービス］　多くの例で、文書デリバリーアプリケーションは、文書の各々の送信者１６ないし受信者２２が保守されねばならないビジネスの用命に応じる。１０万人の受信者２２からなる同じグループに定期的に文書を送付する場合を想定する。文書の送信者１６は、多数の申し込み加入／配布ベースを更新かつ操作するツールを必要とする。ポータブル文書デリバリーシステム１６０は、発行者１６が、ＢＦＤサーバー１２にアカウントを作成し、処理を特定のアカウント１３２、１３４、１３６に関連付けることを可能にする。システムは更に、発行者が、多数のユーザーアカウントを単一ビリングアカウント１３４に統合することを可能にする。加えて、発行者が、トランザクションに特定ビリングコードを関連付けさせるのを可能にし、トランザクションが処理レポートで統合されるようにできる。例えば、法律事務所は、アカウント、次に各クライアントのビリングコードを作り、ビリングコードとアカウントを各文書トランザクションに関連付ける。ポータブル文書デリバリーシステム１６０は、アカウント情報を自動的に維持・更新する。ポータブル文書デリバリーシステム１６０の報告機能は、ユーザーが所定のアカウントないし特定のビリングコードに関する報告を作成するのを可能にする。こうしたスキームは、ビリングと同様にクライアント管理を容易にする。

【００４８】［トランザクションマネジメントサービス］　トランザクション管理の要件はアカウント管理に関連する。文書の送信者１６と受信者２２のデータベースを維持することのみならず、送信文書の処理を管理するサービスを提供することも必要である。例えば、送信者１６は、文書が実際に送付され実際に受信されたか、そして恐らく誰が文書を受信したかを知りたい。多くの例で、発行者１６は、送付の郵便料金を請求したいであろうし、それ故に送付処理に関連する会計情報を維持・更新するサービスを必要とする。ポータブル文書デリバリーシステム１６０は、各送信処理に関連した記録を作成でき、これら記録を維持できる。各トランザクションないし文書送信操作は、特定のアカウントに関連している。ユーザー１６は、サーバーから処理情報を直接照会できる。

【００４９】［レポーティング］　アカウントとトランザクションの管理は、レポーティングの精巧な手段が提供されなければ何ら価値がない。例えばユーザー１６は、特定の文書が誰に送付されたか、何人のユーザーが文書の送付を確認したか、請求を目的としたトランザクションに関する費用といった情報を含む所定のトランザクションの完全なレポートを提供してもらえる。

［規模と帯域幅］　文書デリバリーアプリケーションの領域と応用は広い故、ポータブル文書デリバリーシステム１６０は、無数の文書ないし受信者２２にサービス提供する性能を拡張できる。送付プロセスでの幾つかの局面はリアルタイムで起こる一方、他の局面は延期されるないしスケジュール化される。多くの場合ポータブル文書デリバリーシステム１６０は、帯域幅の量又は展開されるサーバー１２ａ－ｎのセットを動的に広げて、文書送付に必要な処理量を達成する。

【００５０】ポータブル文書デリバリーシステム１６０は、ユーザーの要求を満たすようにスケールが変われる。サーバーソフトウエアは、一日に無数の文書を送信するのを支援するよう設計されており、所定のサーバーに専用となったどんな帯域幅でも活用できる。例えば、最近のＢＦＤサーバー１２は、１０メガビット／秒の帯域幅を効率的に利用する。ＢＦＤサーバー１２で作動する様々なプロセスは、非同期で作動して、多重処理サーバー１２での最適な作動を可能にし、同様に所定のトランザクションサービスを精巧にスケジューリングすることも可能にする。リアルタイムの作動には特に注意が払われ、受信者２２がサーバー１２から文書へアクセスすることに関しては特別の注意が払われる。ＢＦＤサーバー１２は、他のサーバー１２ａ－ｎの間での作業負荷を配分することもできる。本発明の好適実施例では、単一のサーバー１２で作動している個々のプロセスを一群のサーバー１２ａ－ｎに配分されるのを可能にする。本実施例では、アカウント管理プロセスを一つのサーバー（例えば、１２ｄ）上で作動させる一方、ロギング、レポーティング、トランザクション管理、送信、宣伝及び検索の諸プロセスを別のサーバー（例えば１２ｈ）上で作動させることが可能である。

【００５１】［ポータブル文書送信クライアント明細書］　ポータブル文書送信クライアント（ＰＤＳＣ）１９２は、どんなコンピューターユーザーでも、パソコンないしマッキントッシュコンピューターといったどんなパーソナルコンピューターのデスクトップから直接、文書を配布できるようにする。ＰＤＳＣ１９２は、仮想のプリンター装置を使い全てのアプリケーション１９０を直接組み込むので、ＰＤＳＣ１９２は、全てのアプリケーション１９２とフォーマットに対し互換性を持てる。重要なのは、ＰＤＳＣ１９２が直接ポータブル文書技術に組み込まれているので、文書の送信者１６は、文書の対象受信者２２の能力について仮定を設ける必要がないことである。ＰＤＳＣ１９２で、使用法の二つの主要モード：プリントないし「ドラッグアンドドロップ」が可能になる。プリントのモードで、送信者１６は、アプリケーション１９０からプリントオプションを単純に選択し、ポータブル文書を作成するイベントシークエンスを開始させ、その後文書をアドレスし送信できる。ユーザーからみると、ユーザーはプリントコマンドを単純に選

び、標準アドレシングインターフェースとアドレスブックを使い、文書の最終目的地を指示するよう促される。例えば、マイクロソフトメール™のユーザーは、標準マイクロソフトメール™のアドレシングダイアローグで文書をどこへ送信するか命ずるよう促される。ＰＤＳＣ１９２は、文書の最終目的地を選んだ後、ＢＦＤサーバー１２に自動的に接続し、送信をカスタマイズするために選ばれた他の属性と同様に、文書１６６と対象受信者２２のリストとを安全にアップロードする。「ドラッグアンドドロップ」使用法では、ユーザー１６は文書を送信するためのアプリケーションの開始と印刷を無しで済ませられる。文書は、ＰＤＳＣ１９２送信アイコンに単純にドロップされ、送信者のデスクトップ１６４からアクセスできる。

【００５２】追加の機能性とカスタマゼーションは、一回のクリックアウェイである。アドレシングプロセスの間、ユーザー１６は、高級オプションを呼び出して、送信オプションを自由にカスタマイズできる。デフォルトにより、各送信では既存のパラメーターを再使用し文書を送信することになる。ユーザー１６は高級オプションユーザーインターフェース１９３を使って、例えば機密保護オプション及びレシート要求を含むデリバリーオプションをカスタマイズすることもできる。例えば、プライベート及び／又はパブリックキー暗号化を含む機密保護オプションをユーザー１６がカスタマイズしたければ、ユーザーは「パブリック暗号化」ないし「プライベート暗号化」のオプションをチェックするだけである。同様に、ユーザーは「レシート通知」オプションを選択でき、こうして何時文書が実際に受け取られたか受信を確認する様ＢＦＤサーバー１２に告げる。

【００５３】［ＢＦＤ　サーバー構成オプションユーザーインターフェース］　ＢＦＤサーバー１２を、本来送信者デスクトップ１６４から構成しカスタマイズすることが可能である。デスクトップからＢＦＤサーバー１２へのアクセスは、ＨＴＭＬ型ユーザーインターフェースを使って達成される。このユーザーインターフェースは、ＢＦＤサーバー１２の高級オプションへのアクセスと制御をサーバー管理者に与えるためにある。例えば、サーバー管理者は、特定の文書を受け取る予定の１０万人の受信者のデータベースを更新し、これら受信者への文書送信を直接呼び出してもよい。サーバー管理者は、先週起きた送信処理に関してのレポートを作成できる。デスクトップ１６６からＢＦＤサーバー１２にアクセスするには、ユーザー１６は、ＢＦＤサーバー１２上に作成された特別のアカウントを持たねばならず、これをＢＦＤサーバー１２が前もって作成する。加えて、このアカウントの上側のＢＦＤサーバー１２にアクセスするには、認証及び機密保護の幾つかの層を必要とするので、こうして一方的なアクセスを防ぐ。

【００５４】サーバー構成ユーザーインターフェース１９８は、ユーザーが、サーバーセッティングにアクセスし制御するのを可能にし、これにはトランザクション管理、アカウント管理、レポート便宜、配布文書の直接アップロードとダウンロード、受信者リストの直接操作、及び送信オプションへの直接アクセスが含まれる。［ポータブル文書受信クライアント］　文書の受信者２２は、ポータブル文書受信クライアント（ＰＤＲＣ）１９４を利用して、ＢＦＤサーバーアドミニストレータを直接経由してポータブル文書送信クライアント１９２ないしＢＦＤサーバー１２により受信者２２に送られた文書に、アクセスし手を加えることができる。文書の受信者２２がＰＤＲＣ１９４をまだ所有していない場合、ソフトウエアはインターネットから直接ダウンロードされて設置されてもよい。ポータブル文書デリバリーシステム１６０のアーキテクチャは、このプロセスを簡素化し、専用ソフトウエアとスクリプトを使用するが、加えて初心者である受信者２２でも文書受信に必要なソフトウエアへのアクセスを一回のクリック操作で可能にする新しいブラウザアーキテクチャを出現させた。

【００５５】ポータブル文書受信クライアント１９４の最も基本的なケースは、ネットスケープナビゲーター™のプラグインないしマイクロソフトアクティブＸ™コントロールといったブラウザ拡張子として単純に機能できることである。他のユーザーにとっては、ＰＤＲＣ１９４は、ヘルパーアプリケーションとして作動する分散型アプリケーションとして機能する。第三のアプリケーションは、受信者のデスクトップ１７０からポータブル文書へ直接アクセスすることを望むポータブル文書デリバリーシステム１６０の顧客のためにある。この構成では、専用ポータブル文書受信クライアント１９４をインターネットから直接ダウンロードできる。このコンポーネントは、ポータブル文書デリバリーシステム１６０の作動を連続的に監視し、入ってくるどんなポータブル文書をもＢＦＤサーバー１２から自動的に引き抜いて、受信者２２のコンピューターデスクトップ１７０上で直ちに文書通信として開く。

【００５６】ポータブル文書デリバリーシステム１６０からのポータブル文書の受信者２２は、送信構成オプション次第であるが、文書からの情報を見る、探す、プリントする、保管する、送り出すことができる。ポータブル文書デリバリーシステム１６０に関連してＥｎｖｏｙ™ないしアクロバット™を使って配布された文書は、完全な視覚忠実性を保ち、最高レベルの品質と解像度で高解像度出力装置上に作成できる。図２０は、ＢＦＤサーバー１２のファックスゲートウエイ５６によりプリンター１７８へ文書がどう送られるかを示す。図２１は、専用共同ＢＦＤサーバー２００のデパートメントゲートウエイ２０２によりＬＡＮ２０４を通って、文書がどう送られるかを示す。

【００５７】［指示された文書送信のためのプライベー

トかつ追跡可能なＵＲＬｓ］本発明のこの実施例は、文書を電子的に送付する独自の方法を提供している。重要なのは、本発明のこの実施例が、基本的な文書送信に加えて、トラッキングと機密保護を含みはするがこれらのみに限定されない付加価値のある多くのサービスを可能にしていることである。本発明は、情報を配布するためにプライベートなユニフォームリソースロケータ（ＵＲＬ）を動的に作成する文書送付アーキテクチャを提供する。各プライベートＵＲＬ（「ＰＵＲＬ」）は、文書の対象受信者、送付される文書ないし文書セット、送付プロセスに固有な他の（随意な）パラメーターを独自に定める。文書の対象受信者はＰＵＲＬを使い文書（ないし複数の文書）を検索する。サーバーは文書の検索の際、データベース中の検索に関連するログ情報と同様に、ＰＵＲＬに含まれている属性に基いた検索の行動をカスタマイズする。ＰＵＲＬｓのアーキテクチャと使用法が、文書の安全な送付と文書受信の追跡を可能にする。

【００５８】ワールドワイドウエブ（「ウエブ」）は、顧客がウエブブラウザを使用してウエブサーバーから内容を検索できるようにする。つまり、顧客はウエブから内容を引き出す。Ｅ－メールは、内容の作成者がその内容を顧客に送ることを可能にする。言い換えれば、作成者は内容をＥ－メールでプッシュする。Ｅ－メールインターネットサーバーは、インターネットサーバーの行動を統制するＳＭＴＰプロトコル（単純化されたメールトランスポートプロトコル）同様に、インターネットのユーザーに提供する限られた能力である。例えば、ＳＭＴＰのＥ－メールサーバーは、バイナリーファイルのタイプ、トラッキング、機密保護について何も知らない。ウエブ及び関連したＨＴＴＰプロトコルは対照的に、バイナリー情報の効率的かつ安全な送信を可能にする柔軟なプロトコルを提供する。しかしＨＴＴＰは、プル型顧客作動のプロトコルである故、情報の作成者ないし送信者は、情報送付を指示するのにＨＴＴＰだけを頼りとすることができない。

【００５９】送付のためのＨＴＴＰを組み合わせ、同様に通知のためのＳＭＴＰ／Ｅ－メールを使うことにより、作成者がドライバーになるないしプッシュするのを可能にし、更にＳＭＴＰ／Ｅ－メールに関連した限定条件や生来の問題とは無縁な解決策を作ることが可能である。ＰＵＲＬｓは一時的かつ動的に作られたＰＵＲであり、文書の送付に関連する属性と同様に、文書の対象受信者と文書自体を独自に確認する。ＰＵＲＬｓは、文書を送信するためＥ－メールメッセージに情報を付けるのを防止し、逆に送られる文書に汎用参照番号をつけ、受信者が参照番号を経て文書にアクセスするのを可能にする。受信者が参照番号を使って文書にアクセスすると、サーバーは、文書にアクセスする要求を傍受し、トラッキング・機密保護といった付加価値のあるサービスを提供する。例えば、ユーザーは、サーバー上で文書のロックを解除する役を果たす、即ち、暗号化された文書を恐らく解読するキーをＰＵＲＬに含ませることができる。ないしユーザーは、受信者を確認する独自の身分証明番号をＰＵＲＬに含ませることができる。この場合、サーバーは、特定の個人が特定の文書にアクセスしたことに注目でき、それをデータベースに書き留め、送信者にその情報を利用できるようにすることができる。従って本発明の本実施例は文書追跡を可能にする。

【００６０】図２２は、本発明により命じられた文書送付のためのプライベートかつ追跡可能なＵＲＬｓを含む文書デリバリーシステムを示すブロック図である。文書３１０は送信者３００からサーバー３１５に送られる。サーバーは文書を一時的に記憶する。サーバーは、文書の各対象受信者のためにＵＲＬを動的に作成する。ユーザー情報と文書情報をＵＲＬで暗号化するのに加え、サーバーは、デリバリーパラメーター或いはＵＲＬのトランザクション識別子も暗号化する。生成された各々の個人ＵＲＬ（ＰＵＲＬ）は、次に各対象受信者３２０に送られる。特定の文書が送られてきた旨の通知３２５が受信者に出される。通知は、プライベートなＵＲＬを含むＥ－メールメッセージの形式が代表的である。受信者はＰＵＲＬ３３０とウエブを使って文書にアクセスする。

【００６１】受信者がＰＵＲＬを経て文書にアクセスする時、受信者はサーバーにＰＵＲＬを示す。この時サーバーには、次の一連の動作を決める機会がある。例えば、サーバーは、パスワードを示してからでないとＰＵＲＬが言及した電子文書にアクセスできないとＰＵＲＬに記されていることに気付く。サーバーは、又ＰＵＲＬで文書にアクセスする特定の受信者を確認し、特定の受信者が特定の文書にアクセスを試みたことを記録し、全てがＰＵＲＬによって再度確認される。サーバーは、更に全ての文書が首尾よく送付された事実を記録する。それ故、サーバーに維持されているデータベースは以下に述べる記録を完全に有しており、例えば次の通りである。

- 誰が文書にアクセスしたか；
- いつ文書にアクセスしたか；
- 首尾よく文書にアクセスしたかどうか。

【００６２】サーバーが記録したこの情報は、文書の送信者に報告される。従って、通知のためのＥ－メール、送付のためのウエブ、受信者と文書を確認するためのプライベートＵＲＬｓの組み合わせを使えば、文書をトラッキングし、送信者に文書の送付状態を報告する様に、デリバリーサーバーを構成することが可能である。そうしたシステムを実際に実行するには、図３－２１との関係でここに述べられているシステムに従ってもよく、又適切な他の形式でもよい。本発明の他の実施例で、サーバーは他のタイプの情報を記録できる。こうしてサーバーは、文書を検索する特定の受信者に関するＩＰアドレスを記録できる。サーバーは更に、同じＰＵＲＬのつい

た特定の文書へ引き続きなされるどんなアクセスのＩＰアドレスをも記録できる。このように、サーバーは、多数のＩＰが同じキーを使用して同じ文書にアクセスするのを防ぐ。代替案として、サーバーは、特定の受信者を対象とした特定の文書にアクセスしたＩＰアドレスを含むリストを送信者に提供することも可能である。

【００６３】送付のための上記のアーキテクチャは、機密保護も容易にする。文書にアクセスし解読するための有効キーを受信者が示すまで、サーバーで文書を暗号化されたままにできる。このキーは暗号化されて、ＰＵＲＬの一部に示される。代替案として、キーを検索せねばならないとＰＵＲＬが述べており、この場合サーバーは、文書を解読するための独自のパスワードを示すよう受信者に要求する。第一の場合、キーはＰＵＲＬのカプセルに包まれているので、暗号化された文書の検索は第一段階の自動プロセスである。

【００６４】［ＰＵＲＬ実行］最初にＰＵＲＬの潜在的構造を考える。次の図で、ＰＵＲＬの特定の一例を略述する。

http://posta.tumbleweed.com/cgi/posta.dll?pu=0-233-33982-FIAAAV4

上記ＰＵＲＬの意味は、次の通りである。

意味	バリュー
アクセスにＨＴＴＰプロトコルを使用せよ	http：／
ＨＴＴＰサーバーの名前	posta.tumbleweed.com
ＨＴＴＰサーバー拡張子の名前	cpi/posta.dll
パスワード使用禁止	pu=0
ストア項目識別子	２３３
受信者識別子	３３９８２
文書にアクセスするためのキー	FIAAAV4

更に図２２を参照し、ＰＵＲＬ３０２が様々なフィールドを有しているのが示されていることに注目されたい。これらのフィールドには、パスワード識別子３３１、ストア項目識別子３３２、受信者識別子３３３、文書キー３３４、希望する他のオプションフィールド３３５が含まれる。これらフィールドをより詳しく以下に述べる。

【００６５】［パスワード識別子］　パスワード識別子は、所定の文書にアクセスするためにパスワードが要求されているかどうかを記す。この場合、バリュー“０”は、パスワードが要求されていないことを示す。バリュー“１”は、パスワードが要求されていることを示す。

［ストア項目識別子］　ストア項目識別子は、所定の受信者がどの文書を欲しがっているかを独自に定める。この場合、バリュー“２３３”は、サーバー上のまばらな表に索引を提供し、例えば、所定の文書がサーバーのどこにあるか及び／ないし文書が何と名付けられているかを確認するバリューを定める。

［受信者識別子］受信者識別子は、所定の文書の対象受信者を独自に定める。この場合、バリュー「３３９８２」は、サーバー上のまばらな表に索引を提供する。この表索引にあるバリューは、受信者情報を含む。

【００６６】［文書キー］　文書キーは、ＰＵＲＬ自体を確認する。この場合、キーは、所定の受信者とストア識別子に関連して無作為に作られた数字である。キーは、所定の受信者識別番号が有効か、所定のストア識別番号が有効か、及び所定のストア識別番号を有する所定の受信者の文書へのアクセスが承認されるかどうかを確認するために使われる。本発明の他の実施例では、キーが、確認情報を含む表の中へ索引をも暗号化しており、確認情報それ自体を暗号化するのとは対照的である。重要なのは、サーバーが、ウエブ拡張子を有し、カスタマイゼイションを提供するように文書のＨＴＴＰプロセスが拡張されるのを可能としている点である。こうして、文書にアクセスする受信者は、ＨＴＴＰサーバー拡張子を通してＨＴＴＰサーバーと通信する。本拡張子は例えば、文書にアクセスする承認を決定でき、その場合、特定文書の送信を容易にする新しいＰＵＲＬをユーザーに示す。

【００６７】サーバーは、ＰＵＲＬの上記属性とバリューを使って、文書送付の挙動をカスタマイズできる。特定すれば、サーバーは、文書を送付し、送付トランザクションを記録するため以下のステップを実行する。

・ＰＵＲＬを様々な部分に暗号化する。
・ＰＵＲＬの各コンポーネントを確認する。
・キーで、ＰＵＲＬを認証する。
・受信者識別子で、どのユーザーが文書にアクセスしているかを決定する。
・ストア項目識別子で、どの文書にユーザーがアクセスしているかを決定する。
・上記を前提に、文書が送付される前に追加入力を要求するか否かを決定する。
・文書を受信者に送付する。
・送信の全属性、例えば、アクセス時間、送信の成功、受信者のＩＰ、を記録する。

【００６８】ひとたび情報が、トランザクション情報を記録するサーバー上で作動するデータベースに記録されると、このデータは、受信者によってアクセスされ、受信者に動的に送信されて戻る。例えば、所定の発行者（送信者）は、特定の受信者に送付された全文書に関するサーバーのデータベースを照会できる。発行者は、１０人に送られた所定の文書の状態レポートを作成するようサーバーに照会する。サーバーは、例えば、文書が特

定の時刻に１０人全てに送られたが、３人しか実際に文書を検索しなかったと報告する。各文書の検索には、文書がアクセスされた特定時刻、アクセスされた時間、全部が首尾よくアクセスされたか否かが含まれる。それ故、動的に作成されＥ－メールで広められるＰＵＲＬは、広範なネットワークでの文書送付を追跡する強力な手段をもたらす。

【００６９】電子文書デリバリーシステムとその使用方法を、インターネット中での使用と関連させてここに述べたが、本発明は、インターネット、イントラネット、ＬＡＮとＷＡＮ、ないし望む所のそれらのどの組み合わせをも含む多様なネットワークのどれにでも適用できる。更に本発明は、多様なコンピュータープラットフォーム、通信プロトコル、ポータブル文書フォーマットないし望まれるそれらのどの組み合わせにも適用できる。本発明は、広範なネットワークでの安全な文書送付のための方法とシステムを提供する。文書デリバリーサーバーが、文書の対象受信者のパブリックキーを動的に検索した後、パブリックキーを使って文書ないし文書のシークレットキーを暗号化する。サーバーは、広範なネットワーク、例えばインターネットで対象受信者に暗号化された文書を送付する。対象受信者は、パブリックキーに関連するプライベートキーを使い文書を解読する。本発明は、対象受信者にのみ特定の文書へのアクセスを許し、それ故文書送付のために独自のレベルの機密保護を提供する。

【００７０】本発明の目的のため、文書なる用語は、同一限界内のデータ集合なら全てを含んでおり、データストリーム、ビデオ、音声データ、アニメーション、ＨＴＭＬ・ＰＤＦ・Ｅｎｖｏｙといったフォーマット化された文書、ないしデータベースが含まれる。本発明の好適実施例がインターネットでの文書送信の使用に適応している一方、本発明は、他の広範なネットワークにも同等に適用できる。更に、本発明の好適実施例が受信者コンピューターへの文書送信を開示する一方、本発明は、プライベートキー／パブリックキーを動的に生成しかつプライベートキーを使って対応するパブリックキーで暗号化された文書を解読する能力を維持ないし有するどんな対象受信者に対して、文書送信を作動させることができる。それ故対象受信者として、例えば、デスクトップコンピューター、ファックス機、パーソナルデジタルアシスタントないしネットワークコンピューター装置のインターネットユーザーが含まれる。

【００７１】同様に、文書送信者はデスクトップコンピューターであるのが好ましい一方、送信者として、ネットワークコンピューター装置の様な文書暗号化できデリバリーサーバーと通信できる装置が含まれる。本発明の代替実施例では、文書をデリバリーサーバーにより暗号化する。この実施例では、送信者として、インターネットブラウザ装置、インターネット電話装置、パーソナルデジタルアシスタントないしファックス機といった、暗号化と対象受信者への送信のためにデリバリーサーバーへ文書を伝送できる装置なら何でも含まれる。図２３は、本発明の第一好適実施例による動的サーバー文書暗号化システムの図である。デスクトップコンピューターに記憶された文書を送信者１０３２は、もう一つのコンピューター、対象受信者１０３４に送信する。この第一好適実施例で文書は、ポータブル文書フォーマット（ＰＤＦ）で記憶される。しかし代替実施例では、文書は、どの適切なフォーマットで記憶されてもよい。ポータブル文書（ＰＤ）のフォーマットが、配布されたプリントとファックス解決策に必要となる。しかし、本発明にＰＤフォーマットは必要でない。

【００７２】文書は、デリバリーサーバー１０３６を経て送信者から受信者に送られる。本発明のこの第一好適実施例の場合、デリバリーサーバーは、対象受信者のパブリックキー（証明）を検索するために証明許可データベースサーバー１０３８と通信するよう、送信者により指示される。デリバリーサーバーは、証明元へ動的に照会し、パブリックキーを検索する。パブリックキーはデリバリーサーバーへ送信され、そこから送信者に送信される。本発明の代替実施例において、デリバリーサーバーは、対象受信者のデスクトップコンピューター、インターネットサーバー、ないし対象受信者のデスクトップコンピューターに接続されたイントラネットサーバーから、対象受信者のパブリックキーを検索する。本発明の第一好適実施例において、送信者は、シークレットキーを使用して文書を暗号化し、パブリックキーを使ってシークレットキーを暗号化する。文書と暗号化されたシークレットキーは、次に対象受信者に送信される。シークレットキーが、対象受信者のプライベートキーで解読された後、文書を解読するのに使用される。

【００７３】代替の同等好適実施例として、送信者は、パブリックキーを使って文書を暗号化する。暗号化された文書は対象受信者に送信され、パブリックキーに関連したプライベートキーを使って解読される。図２４は、本発明の第一好適実施例による動的サーバー文書暗号化のための操作セットのフローチャートである。本例では、１０４０のステップで、送信者はシークレットキーを使って文書を暗号化する。こうしたシークレットキーには、先行技術で知られているどんな適切な暗号化スキームも含まれる。１０４５のステップで、送信者はデリバリーサーバーと交信し、１０５０のステップで、対象受信者に関連したパブリックキーについてに照会する。１０５５のステップで、デリバリーサーバーは、例えばこの証明をリアルタイムで証明元のデータベースから検索し、１０６０のステップで、送信者に証明を送信する。

【００７４】最終的に証明元が証明を返してこない場合、デリバリーサーバーが、受信者に新しい証明を動的

に生成する。こうすることで、デリバリーサーバーは、動的に作られたＵＲＬをＥ－メールメッセージに入れ受信者へ送る。受信者がＵＲＬにアクセスすると、ジャバアプレットないしプラグインが検索され、受信者のシステムに自動的にダウンロードされる。このアプレットないしプラグインは次に受信者システム上で作動し、一対のプライベート／パブリックキーを作成する。一対のプライベート／パブリックキーをローカルマシンで生成することは、本発明に固有なことではなく、多くの出典で述べられている。アプレットないしプラグインは、次にパブリックキーをデリバリーサーバーへ送る。サーバーは、作成されたＵＲＬのプロパティを使い、受信者のＥ－メールアドレスを確認する。こうして、生成されたパブリックキーは、受信者の電子メールアドレスを認証済みとしたプロパティを有する、何故ならキー生成を呼び出すＵＲＬは特定の電子メールアドレスに送られただけだから。サーバーは、電子メールアドレスとパブリックキーを結合させて証明とし、この証明は送信クライアントへ戻されるか、ないし、文書又はシークレットキーを暗号化するためにサーバーにより使用される。デリバリーサーバーは、ＬＤＡＰないし類似のプロトコルを使って、証明を証明元に伝達する。代替案として、ローカルデータベースないし動的に生成された証明を、デリバリーサーバーは将来の使用のために単に維持してもよい。

【００７５】デリバリーサーバーからパブリックキーを受信すると、送信者はパブリックキーでシークレットキー６５を暗号化する。本発明の代替同等好適実施例では、送信者は、パブリックキーを受信するまで文書を暗号化しない。何故ならパブリックキーが認証されなければ文書は暗号化されないからであり、この実施例は、パブリックキーが検索できない時の処理時間を最小にする。次にスッテプ１０７０で、送信者は、暗号化された文書、対象受信者アドレス（例えば電子メールアドレス）、送付指示、暗号化されたシークレットキーを安全なチャンネルで、デリバリーサーバーへ送る。こうして、文書がシークレットキーで暗号化され、シークレットキーが対象受信者のパブリックキーで暗号化されるまで、文書は送信者から離れることがない。次にスッテプ１０７５で、デリバリーサーバーは、暗号化された文書とシークレットキーを対象受信者に送付する。スッテプ１０８０で、対象受信者はパブリックキーに関連したプライベートキーを使ってシークレットキーを解読し、シークレットキーを使って文書を解読する。こうしたスキームは、文書にアクセスできるのはパブリックキーの所有者のみだから、文書への認められていないアクセスを防ぐことになる。

【００７６】図２５は、本発明の代替実施例による、動的サーバー文書暗号化のための操作セットのフローチャートである。ステップ１０９０で、送信者は所定の受信者に文書を送りたい旨をデリバリーサーバーに通知する。ステップ１０９５で、デリバリーサーバーは、対象受信者のパブリックキーを得るべく証明元に照会し、ステップ１１００で、パブリックキーがデリバリーサーバーに戻される。この実施例では、送信者は文書を暗号化せずに、ステップ１１０５で安全なチャンネルでデリバリーサーバーへ文書を送る。次にステップ１１１０で、デリバリーサーバーは、シークレットキーを使い文書を暗号化する。ステップ１１１５で、デリバリーサーバーは、対象受信者から検索したパブリックキーを使ってシークレットキーを暗号化し、次のステップ１１２０で、暗号化された文書とシークレットキーを対象受信者に送る。ステップ１１２５で、対象受信者はプライベートキーを使ってシークレットを解読後、シークレットキーを使い文書を解読する。

【００７７】代替案として、デリバリーサーバーは、パブリックキーを使って文書を暗号化してもよい。暗号化された文書は、次に受信者に送信される。本発明の好適実行例では、送信者は、広範なネットワーク例えばインターネットで作動する全てのデリバリーサーバーを経由して、対象受信者に接続される。送信者は、本文中でセンドクライアントと呼ぶソフトウエアを使用したコンピューターであるのが望ましい。デリバリーサーバーは、所定の受信者のパブリックキーを決定し、そのキーをセンドクライアントに送る責任を果たす。デリバリーサーバーは更に、暗号化された文書とシークレットキーを対象受信者に送付する責任を果たす。センドクライアントは、送付される文書、全デリバリーパラメーター、文書を受信する対象受信者セットをまず識別することで、デリバリートランザクションを始める。デリバリーパラメーターは、予定送付時間、機密保護オプション、送付の緊急性、送付のための提示パラメーター及び受取通知といったオプションを含む。

【００７８】センドクライアントは次にデリバリーサーバーと対話を開始し、シークレットキーで文書を暗号化する。対話と暗号化の段階は、送信者のハードウエア及びソフトウエア構成次第で、同時ないし順番で実行される。対話において、センドクライアントは、所定文書の対象受信者にデリバリーサーバーを送る。ひとたびパブリックキーが取得されると、センドクライアントは、デリバリーサーバーが送信クライアントと交信するよう要求する。センドクライアントは、所定の文書の対象受信者の身元を様々な方法で表現する。本発明の好適実施例において、センドクライアントは、対象受信者の電子メールアドレスを対象受信者の識別子として使用する。しかし、センドクライアントは、代替識別子、例えば運転免許証番号、社会保障番号、抽象識別子、シンボルネームないしファックス番号で、対象受信者を確認できる。

【００７９】デリバリーサーバーは、幾つか技術を使って対象受信者の証明を得る。本発明の好適実施例においては、デリバリーサーバーは、証明元のデータベースサ

ーバーと交信し、対象受信者を確認する情報を示し、対象受信者のパブリックキーを要求する。それ故本発明を使用し、リアルタイムで（照会される）プログラムインターフェースを通して動的にアクセスできるパブリックキーデータベースを維持している証明元から情報を得てもよい。本発明は、ユーザーを介在させずリアルタイムで対象受信者のパブリックキーを照会するデリバリーサーバーに適切などんな手段でも実行される。この様に、特定のプロトコルとパブリックキーデータベースにアクセスする手段は、本発明にとって重要ではない。パブリックキーデータベースへのアクセスには、インターネット技術タスクフォースと合同でミシガン大学が開発したインターネットライトウエイトディレクトリアクセスプロトコル（ＬＤＡＰ）基準を使うのが好ましい。ＬＤＡＰサーバーは、ディレクトリと他のサービスを提供する。ＬＤＡＰプロトコルを使えば、所定のサーバーの照会、サーバーに維持されている情報の電子ネットワークでの検索ができる。ＬＤＡＰサーバーへの直接照会が、標準インターネットプロトコルを使って可能である。本発明の代替実施例では、例えば、ＰＲＣ（遠隔手続き呼び出し）を始めとする様々なコネクティビティプロトコルを持つＳＱＬ照会を使う。

【００８０】証明元のデータベースサーバーとデリバリーサーバーは、同一サーバー、別のサーバーのいずれであっても良い。証明元のデータベースとデリバリーサーバーの両方を同じサーバーに維持することは、証明の汎用データベースにアクセスする必要がない文書送付専用アプリケーションにとって有利である。例えば、会社は、インターネット通信に使われる同一サーバー上に従業員のパブリックキーのデータベースを維持してもよい。それ故同じサーバーが、証明元のデータベースとして又社内の事務所間通信のデリバリーサーバーとして使われる。証明元のデータベースサーバーとデリバリーサーバーが別々である実施例においては、デリバリーサーバーは、最新の照会された証明のキャッシュ或いはローカルコピーを維持してもよい。こうしてキャッシュを使うと、同じ受信者と証明を将来照会する際に時間削減となる。

【００８１】本発明は、一人以上の受信者への文書送付を支援する。複数の受信者には、上記で論議したプロセスを、バッチモードで適用する。対象受信者を整理したリストがデリバリーサーバーに送られ、デリバリーサーバーは対応する証明の整理済みリストを回答する。本発明は、送信者から受信者への複数文書送信に使ってもよい。その場合、単一シークレットキーを使って各文書を暗号化する。ひとたび、個々の受信者のパブリックキーを含む証明をデリバリーサーバーが戻すと、単一シークレットキーは、対象受信者の検索されたパブリックキーで暗号化される。各受信者に対し、センドクライアントは、暗号化されたシークレットキーと暗号化された文書を、対象受信者のアドレスとデリバリーパラメーターと共に、デリバリーサーバーに送る。

【００８２】デリバリーサーバーは次に、組み合わされ暗号化されたシークレットキーと文書を各受信者に送る。受信装置は、レシーブクライアントとして知られているソフトウエアを使用する。レシーブクライアントは、標準インターネットブラウザへのプラグイン同様、ジャバアプレットとして現在実行されている。ジャバは、カリフォルニア州マウンテンビューのサンマイクロシステム社が開発したプログラミング言語である。しかし、レシーブクライアントは、送信されたシークレットキーと文書を受信かつ解読できるプログラミング言語でなら他の言語を使用しても実行できる。レシーブクライアントは、ジャバアプレットとして実行される場合、デリバリーサーバーから対象受信者のシステムへ動的に配布される。レシーブクライアントは、プライベートキーを使ってシークレットキーを解読する。この解読されたシークレットキーを受信者が使い、文書を解読する。

【００８３】この発明の好適実施例において、レシーブクライアントは、ハイパーテキストトラスミションプロトコル（ＨＴＴＰ）、即ち、標準インターネットデリバリープロトコルを使って、デリバリーサーバーからの暗号化されたシークレットキーと文書にアクセスする。しかし、レシーブクライアントは他の適切なプロトコルを使用してデリバリーサーバーにアクセスしてもよい。ＨＴＴＰ使う場合、レシーブクライアントは、文書のアドレスと送付さるべき鍵を含むユニフォームリソースロケータ（ＵＲＬ）を送信される。本発明の好適実施例において、文書とシークレットキーは、単一ファイルないしデータストリーム中にまとめられ、ＨＴＴＰを使用するレシーブクライアントにそのまま手つかずの形で送付される。こうして、レシーブクライアントは最大の柔軟性を与えられ、パッケージを検索してそれを受信者ウエブブラウザから解読する。受信者は、どんなウエブブラウザをもあるいは広範なネットワークで送信されたデータを受信できる他のソフトウエアアプリケーションを使用してよい。

【００８４】本発明を、好適実施例に関連してここに述べているが、当業者は、他の応用が本発明の範囲から外れることなくここで述べられている応用に代り得ることを難なく認めるであろう。センドクライアント、レシーブクライアント、及びデリバリーサーバーソフトウエアのソースコードは、周知のプログラミング技術とハードウエアコンポーネントを使用する当業者なら難なく作れる。加えて、レシーブクライアントとデリバリーサーバーの機能も、集積回路、ＥＥＰＲＯＭといったプログラミング可能な記憶装置を始めとする他の手段によって成し遂げることができる。本発明の好適実施例に関して上記で論議した動的サーバー文書暗号化の実行は、可能な実行の一つにすぎない。代替実施例は、本発明の教示と

一致する他の実行を使ってもよい。

【００８５】レシーブクライアントは、文書を別の装置に向けるように構成してもよい。例えば、解読された文書をプリンターないしファクス機に送信してもよい。本発明では、ＲＳＡ法、ベリサイン法を始めとする適切などんな暗号化スキームをも、シークレットキー、パブリックキー、プライベートキーのために使用してよい。本発明は、特別の好適実施例に関して詳細を述べているが、この発明に関係する当業者は、先の請求項の精神と範囲から離れることなく様々な変更修正と改良ができることを認めるであろう。

【図面の簡単な説明】

【図１】先行技術によるシークレットキー暗号化方式を示す。

【図２】先行技術によるパブリックキー暗号化方式を示す。

【図３】図３は、バイナリーファイルサーバーを一台使うバイナリーファイルデリバリーシステムを示す。

【図４】ファイルサーバーを二台使うバイナリーファイルデリバリーシステムを示す。

【図５】ストア項目のキー要素を示すブロック図である。

【図６】バイナリーファイルデリバリーサーバーの概略図である。

【図７】バイナリーファイルサーバーの一実施例の構造の例を示す。

【図８】バイナリーファイルデリバリーサーバーが使う異なる型のストアイベントを示す。

【図９】バイナリーファイルデリバリーサーバー構造内の特定コンポーネントのブロック図である。

【図１０】ストアの構造を示すブロック図である。

【図１１】インターネットのクライアントをセッション、トランザクション、トランスポートを含む三層にユーザーセッションがどう組織化するかを示す。

【図１２】送信セッションがストア項目を作成した後ないし別のサーバーがストア項目を送っている際の、送信の非対話タスクを示す。

【図１３】アカウントマネージャーの詳細構造を示す。

【図１４】ロガーの詳細構造を提供する。

【図１５】サーバーコネクターの詳細構造を提供する。

【図１６】ポータブル文書デリバリーサーバー一台を使うポータブル文書デリバリーシステムの機能ブロック図を示す。

【図１７】ポータブル文書デリバリーサーバー二台を使うポータブル文書デリバリーシステムの機能ブロック図を示す。

【図１８】ポータブル文書送信クライアントアプリケーションとポータブル文書受信クライアントアプリケーションが、本発明でどう使われるかを示す。

【図１９】サーバーコンフィギュレイションユーザーインターフェースアプリケーションが本発明でどう使われるかを示す。

【図２０】サーバーのファックスゲートウエイにより文書がプリンターへどう送られるかを示す。

【図２１】専用コーポレートサーバーのデパートメントゲートウエイにより文書がＬＡＮを通してデパートメントプリンターへどう送られるかを示す。

【図２２】本発明による指示された文書送付に関するプライベートかつ追跡可能なＵＲＬｓを含む文書デリバリーシステムを示すブロック図である。

【図２３】本発明の第一好適実施例による動的サーバー文書暗号化方法を示す図である。

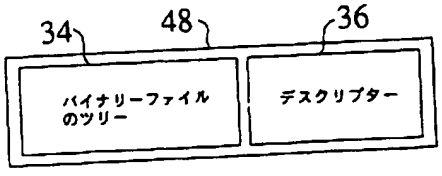
【図２４】本発明の第一好適実施例による動的サーバー文書暗号化のための動作セットのフローチャートである。

【図２５】本発明の代替実施例による動的サーバー文書暗号化のための動作セットのフローチャートである。

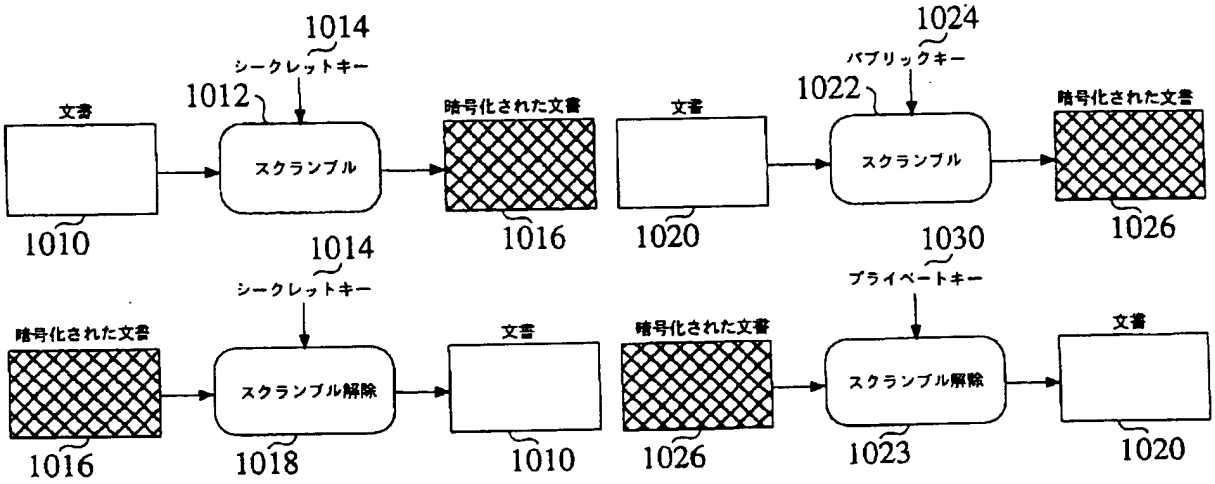
【符号の説明】

３００　送信者
３１０　文書
３１５　サーバー
３２０　受信者（パソコン）
３２５　ＰＵＲＬの通知
３３０　ＰＵＲＬ経由の検索
３０２　ＰＵＲＬ（プライベートユニフォームリソースロケーター）
３３１　パスワード識別子
３３２　ストア項目識別子
３３３　受信者識別子
３３４　文書キー
３３５　オプションフィールド

【図５】



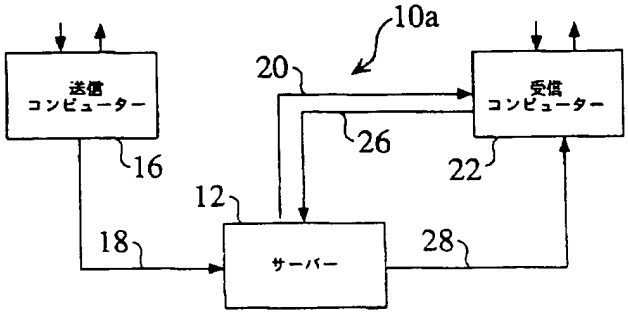
【図1】



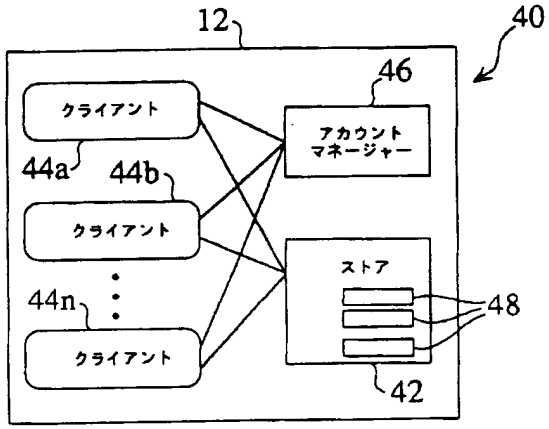
(先行技術)

【図2】

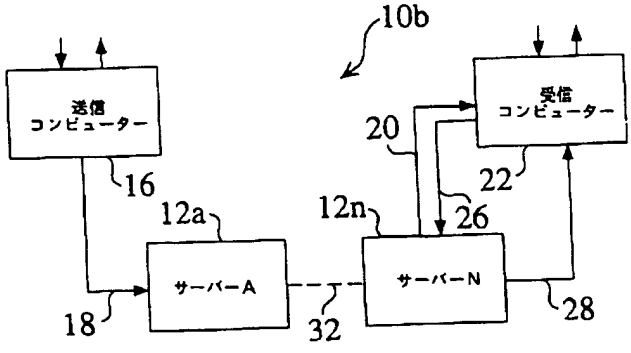
【図3】



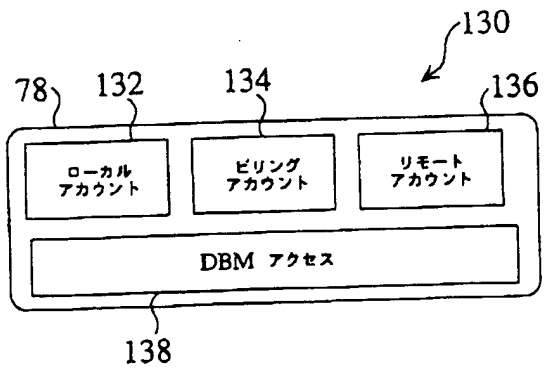
【図6】



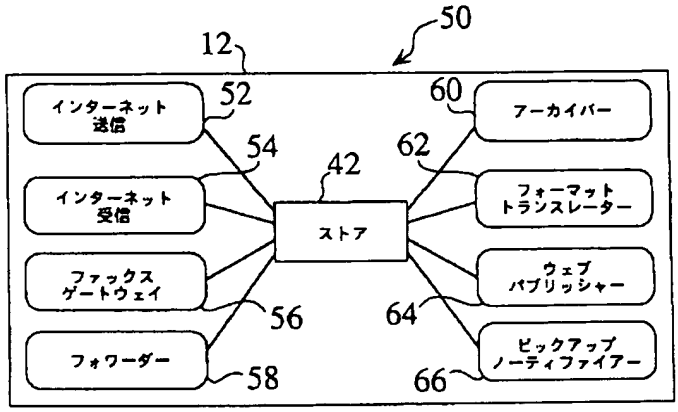
【図4】



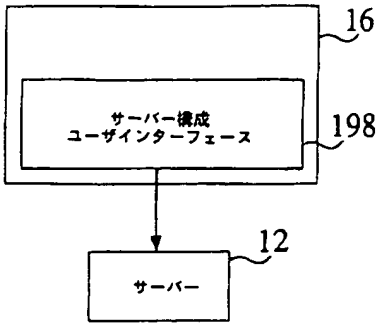
【図13】



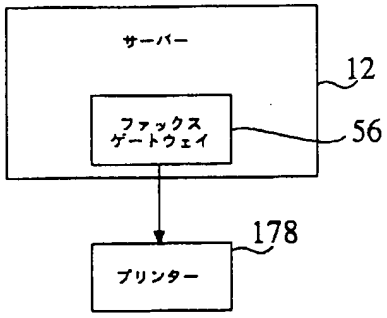
【図７】



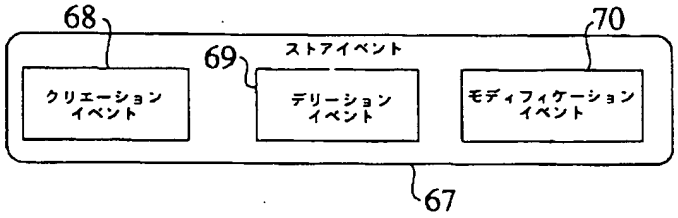
【図１９】



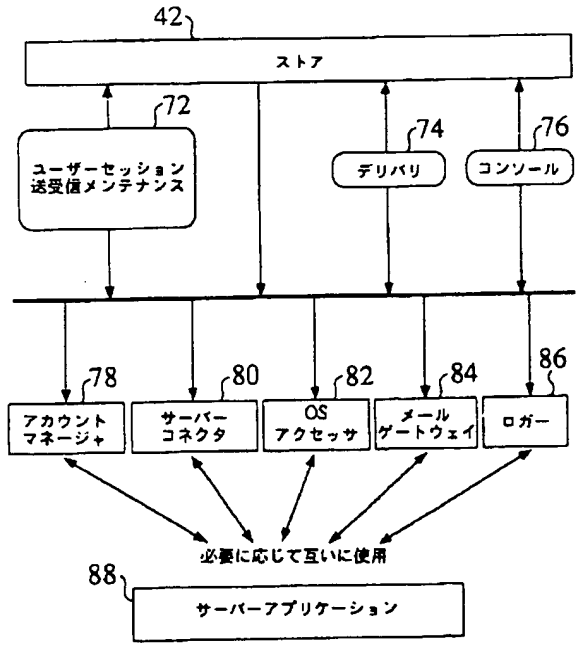
【図２０】



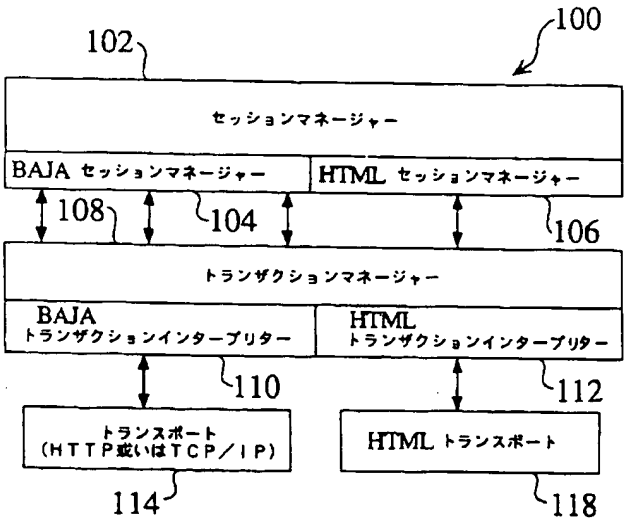
【図８】



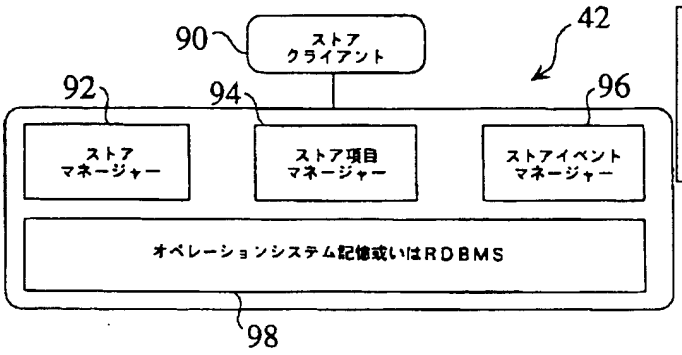
【図９】



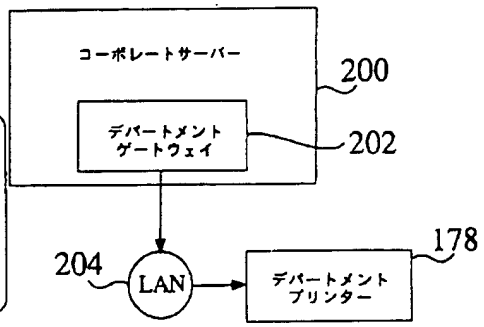
【図１１】



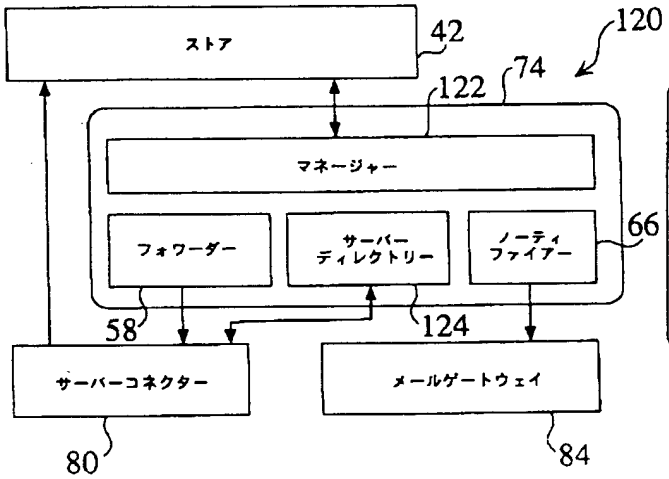
【図１０】



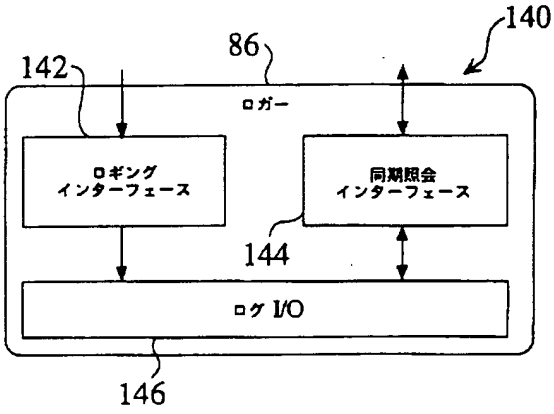
【図２１】



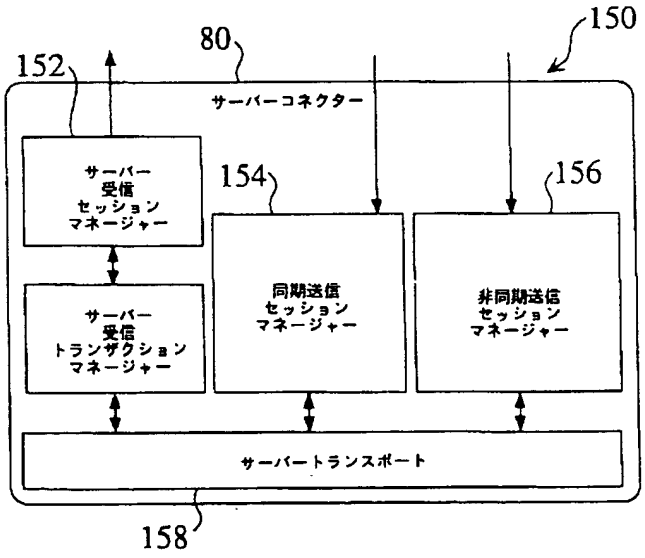
【図１２】



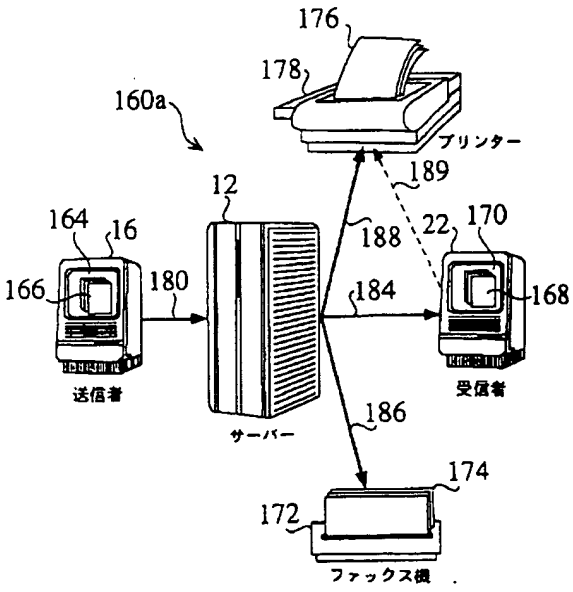
【図１４】



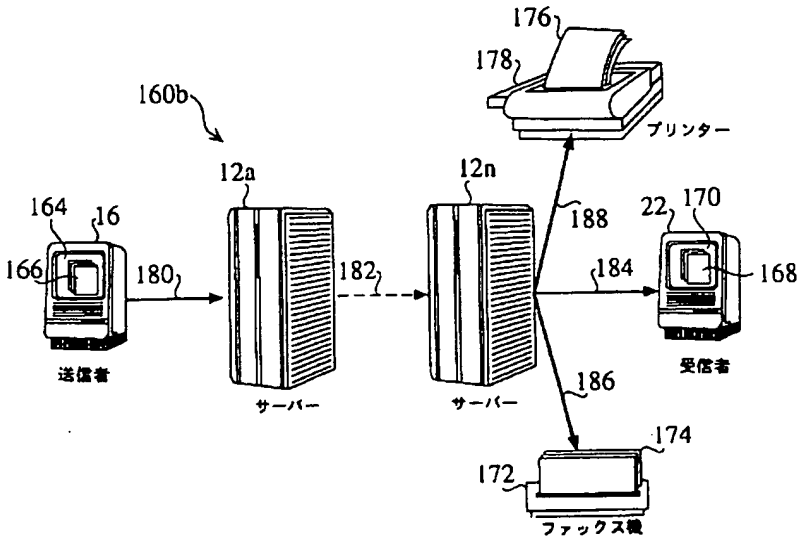
【図１５】



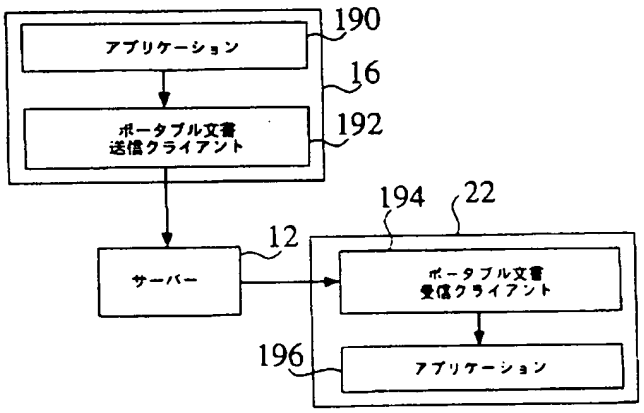
【図１６】



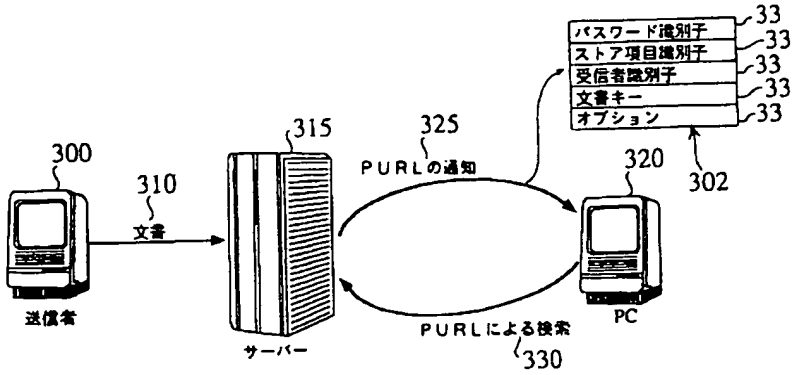
【図１７】


176
178
160b
プリンター
12a
12n
164
16
22
170
188
166
180
182
184
168
送信者
サーバー
サーバー
受信者
186
174
172
ファックス機

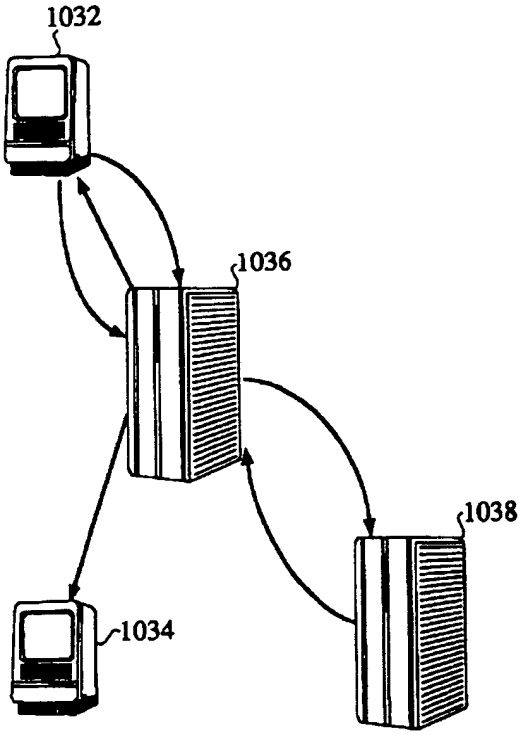
【図１８】


アプリケーション
190
16
ポータブル文書
送信クライアント
192
194
22
12
サーバー
ポータブル文書
受信クライアント
196
アプリケーション

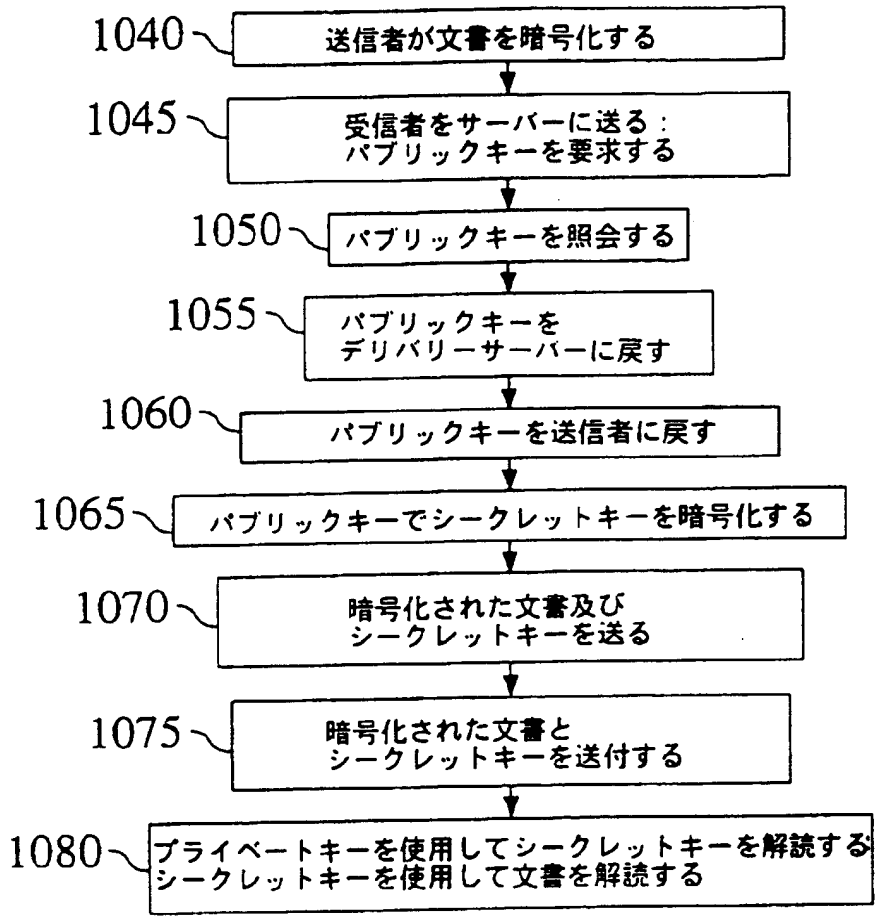
【図２２】


パスワード識別子
ストア項目識別子
受信者識別子
文書キー
オプション
33
33
33
33
33
315
325
320
302
300
310
PURLの通知
文書
送信者
サーバー
PC
PURLによる検索
330

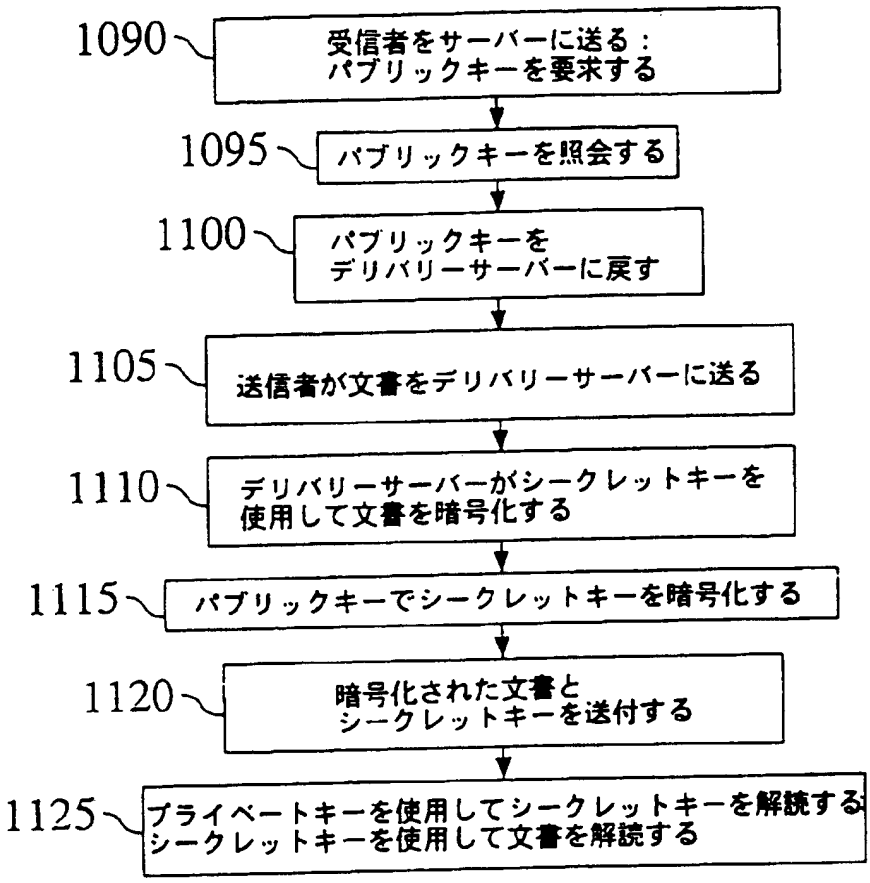
【図２３】



【図２４】


1040
送信者が文書を暗号化する
1045
受信者をサーバーに送る：
パブリックキーを要求する
1050
パブリックキーを照会する
1055
パブリックキーを
デリバリーサーバーに戻す
1060
パブリックキーを送信者に戻す
1065
パブリックキーでシークレットキーを暗号化する
1070
暗号化された文書及び
シークレットキーを送る
1075
暗号化された文書と
シークレットキーを送付する
1080
プライベートキーを使用してシークレットキーを解読する
シークレットキーを使用して文書を解読する

【図２５】



フロントページの続き

(72)発明者 ジャン クリストフ バンディーニ
アメリカ合衆国 カリフォルニア州
95014 クーパーティノ ノース フット
ヒル ブールヴァード 10230 イー10

1) An account manager which maintains all local account information and provides a unique access interface to local accounts, including billing account and remote account information;

2) a server connector 80, which handles all inter-server communications;

3) a mail gateway 84, which handles the sending and receiving of bounced mail;

4) a logger 86, which manages access read/write to the different logs which are classified by a type. The most important log is the send/receive transaction log, which tracks what happens to store items 48 over time; and

5) an operating system accessor 82, which provides a platform independent interface to the operating system for file input and output (I/O), process management (synchronization, locking, threads, process), IPC (RPC, shared memory, shared queues, pipes), network access (TCP/IP sockets, HTTP server interfacing, POP/SMTP interfacing). Specific portions will be implemented as needed.

The Server Application. The server application 88 is used to start up and shut down all pieces of the binary file delivery server 12, according to the configuration parameters. It also provides the administrative aspects of the server not covered by the Account Manager (46 or 78) or by the Logger 86, such as performance profiling, usage information and server parameters/configuration.

Figure 10 provides a block diagram illustrating of the architecture of the store 42. A store manager 92 is used to maintain global state, to synchronize access to the store 42 and to provide housekeeping functions. A store item manager

94 is used to maintain the state, locks, and cache mechanism of a store item 48. A store event manager 96 is used to maintain listener lists and event filters, as well as to dispatch events according to event filters and event priorities.

Figure 11 illustrates how the user session organizes internet clients into three layers, including sessions, transactions, and transports. The session manager 102 maintains all the currently active session states and performs the session-related housekeeping. It processes transactions coming from transaction managers 108 through the uses of the store 42 and the account manager 46. The transaction manager 108 receives raw data from the transport managers 114, 118, and performs validation and preprocessing using one or more BFD transaction interpreters 110 or HTML transaction interpreters 112. The transaction manager 108 then submits the data to the appropriate BFD session manager 104 or HTML session manager 106, waits for an answer, and then passes the answer back to the appropriate transport manager 114 or 118.

Figure 12 illustrates the non-interactive tasks 120 of a delivery, once the send session has created a store item 48 or another server 12 a-n is forwarding a store item 48. The delivery manager 122 listens to relevant store events, makes a forwarding decision, and coordinates work with the notifier 66 and the forwarder 58. The server directory keeps track of the association between E-mail domains and server domains. The notifier 66 is used to handle E-mail notification 20 to the recipient 22. The forwarder 58 is used to forward store items 48 to other servers 12a-n, using a server connector 80. Since not all E-mail notifications may be received, an E-mail scanner is used to check the server mail account for "returned" E-mail, and then to match it with the failed transaction.

Figure 13 provides details of the account manager architecture 130. The account manager 78 is used to maintain user account states 132 for the local

server 12, to maintain billing account states 134 for the local accounts 132, to query local accounts 132, and to maintain a directory of remote accounts 136. The primary goal of the remote account directory 136 is to associate E-mail addresses with either BFD accounts or non-BFD accounts.

Figure 14 provides details of the logger architecture. Figure 15 provides details of the server connector architecture.

System Operation. The following example illustrates how the binary file delivery system 10 is used to distribute electronic information from a sender 16 to a receiver 22. A hypothetical publisher, Sam in Redwood City, California, wishes to send a document to an associate, Rob, in Tokyo, Japan. The following progression of events illustrates how this is achieved, in a controlled fashion.

Sam connects to a local server in Santa Clara, California. Sam's BFD desktop opens a connection to a local server 12a in Santa Clara, where his user account resides. The session manager 102 queries the account manager 78 to validate the user 16 (Sam). The session manager 102 then creates a send session state for the user 16.

Sam's Send Session. Sam's BFD desktop sends transaction details, such as the number of files, file size, and intended recipients. The session manager 102 attaches this data to the session state. Then the session manager 102 creates a store item 48 descriptor 36 in memory, and reserves disk space with the store 42, as well as a store item ID. Then the upload starts. The session manager 102 spools the data directly to a file with asynchronous I/O.

When the upload 18 of all of Sam's files is complete, the session manager 102 updates the store item descriptor 36 to the disk asynchronously, and then inserts the store item 48 asynchronously into the store 42.

The session manager 102 answer's Sam's upload with an acknowledgement, and provides information regarding the transaction. This session then ends.

At the Santa Clara Store. The insertion of the store item 48 is logged asynchronously in the logger 86 by the store 42. The store 42 then runs the store item descriptor 36 against the registered event handlers filters. For each match, it inserts the event and notifiee (Rob) in its event queue. Then that thread dies.

The event dispatch thread pulls the events, and dispatches them asynchronously to the notifiee at rate, depending on the tuning parameters of the system.

The Santa Clara Delivery is Notified. The delivery 74 is notified of a relevant event and starts a thread which waits on the lock of the store item 48 via a synchronous transaction with the store 42. Once the lock is secured, the thread reads the store item descriptor 36, and the delivery manager 74 analyzes it, to decide how to handle it. It turns out the recipient 22 is in the Japan domain, where another BFD server 12n is located. The delivery manager 74 found this out by querying a server directory 124. The manager then decides to forward the store item 48. The forward manager 80 asynchronously asks the Connector 80 to do a forward to Tokyo. Then the thread in the delivery dies. Note that the delivery does not know about the server protocols.

The Santa Clara Connector 80 is going to forward the Tokyo Connector 80. A thread handling the delivery request is eventually started in the Connector 80. It knows the host, and has a lock on the store item 48. It initiates the connection with the Tokyo server 12n. If it cannot connect, it goes to sleep for a while. Eventually, the connection opens, and the connector 80 enters the protocol interpreter, which eventually transfers the store item descriptor 36 and the associated binary data files 34. Then it closes the connection and logs a successful forward to the Tokyo server 12n in the logger 86. Then the connector 80 releases the lock on the store item 48 in the store 42 after having marked it as forwarded.

On release of the lock, the store 42 runs the store item descriptor 36 against the event filter list and finds an event filter that is handled locally. A successfully forwarded store item 48 causes a reference count decreased by 1. In this example, there is only one recipient 22, which means the count goes to zero. Therefore, the store 42 can move the store item 48 to a deletion list. A housekeeping thread of the store 42 will then purge the Store Item 48 at some point.

A thread in the Tokyo connector receiver 80 is begun, to handle the connection. Once the protocol interpreter understands it as a forward, it asks the store 42 for a store item ID 36 and the respective committed storage space. The actual store item descriptor 36 and files have been written to disk as it was receiving the data.

Once the connection is complete, the store item 48 is inserted asynchronously into the store 42 of the Tokyo binary file delivery server 12n.

Tokyo Delivery Component begins. The Tokyo store 42, on insertion, has generated an event which is going to be handled by a thread of the delivery. It

has also logged the insertion of the new item in the logger 86. The manager 102 in delivery 74 realizes this has been forwarded, and that it will be received from this server 12n.

The server 12n queries the account manager 78 to see if there is an account associated with the E-mail address of Rob. If there is no associated account with Rob E-mail, then an E-mail is sent to Rob, with an URL which indicates the store item ID 36. It also queues an asynchronous request for the connector 80 to notify the Santa Clara server 12a that Rob has been notified. If Rob has an account here, then the delivery puts an asynchronous update request with the account manager 78 to mention the pending delivery; in this case the scenario is continued.

Rob connects to the Tokyo Server to check on new documents. When Rob opens its receive session, the session manager 102 synchronously checks the Rob account for validity, and in the process it updates the session state, to remember that the account is flagged with a pending receive. The BFD desktop of Rob eventually asks for the document to be received. The session state has the answer and says yes.

The Rob desktop 170 asks for the receive, and the session manager 102 synchronously asks the store 42 for the lock on the relevant store item 46. Once granted, it can answer by sending the first portion of data. Once the document is downloaded, it asynchronously logs a successful receive with the logger 86. Then it puts an asynchronous request with the connector 80 to notify the Santa Clara server 12a of the final delivery.

At the receive session in Tokyo, the session manager 102 releases the lock, and puts an asynchronous delete request to the store 42. The Rob receive

session then terminates. The connector 80 in Santa Clara runs the protocol interpreter, which says that the notifications must be queued to the logger 86.

Sam checks on Status. Sam connects to do a receive session followed by a maintenance session. The maintenance session 72 receives a request to check on the status of the sent document. The maintenance session 72 synchronously submits a query to the logger 86 using the store item ID 36 that was passed down to the Sam desktop at send time. The query returns the lists of matching records, which are processed and passed back to the desktop, which can then update the user interface 16.

Portable Document Delivery System. Electronic portable documents are becoming increasingly popular. These files can be distributed to different platforms without losing their original look and feel. Adobe Systems' Acrobat PDF™ and Novell's Envoy™ portable document formats have come into widespread use. In a preferred embodiment of the invention, a portable document delivery system 160 achieves a universal solution to the delivery of electronic documents, by applying portable document technology to the Internet. The portable document delivery system 160 provides complete compatibility with portable electronic document formats, including Novell's ENVOY™ and Adobe System's PDF™ formats.

Recipients 22 of portable documents from the portable document delivery system 160 can view, search, print, archive, or export information from their documents. Documents distributed using Envoy™ or Acrobat™ in conjunction with the portable document delivery system 160, preserve complete visual fidelity and may be produced on high resolution output devices with the highest level of quality and resolution. Portable document formats allow preserve content and color of the information within a document, and many formats allow

indexing, searching, and hypertext linking, while allowing the file to be stored in a compact manner.

Figure 14 is a functional block diagram which depicts a portable document delivery system 160a using a binary file delivery server 12. Figure 15 provides a functional block diagram depicting a portable document delivery system 160b using two binary file delivery servers 12a and 12n communicating over the Internet.

To address the limitations of the Web and electronic mail, in addition to providing additional services, the portable document delivery system 160 includes server software which runs on top of existing electronic mail, http server software, and database systems. Thus, the portable document delivery system 160 combines industry standard solutions for the electronic mail, Web, and database to enable corporations and users to direct the delivery of documents to recipients.

The following disclosure elaborates on the requirements for a universal document delivery solution, as well as the specific components of the portable document delivery system 160.

The portable document delivery system 160 combines three basic components to provide a solution to universal document delivery.

1) **Portable Document Send Client.** A portable document send client (PDSC) 192 integrates all desktop applications 190 directly with the portable document delivery system 160. The PDSC 192 is not required for all embodiments of the invention. Publishers who simply wish to leverage the BFD server 12 directly are free to do so. The PDSC 192 is intended for the standard corporate computer user who requires a point-to-point to the delivery problem.

2) **Binary File Server.** The binary file delivery server 12 works on top of Internet standards to deliver documents to recipients. The BFD server 12 can be invoked transparently through the portable document send client (PDSC) 192, or can be invoked and customized directly using a server configuration user interface 198.

3) **Portable Document Receive Client.** The portable document receive client (PDRC) 194 is the software component which recipients 22 of documents utilize to receive, view, and print documents. Recipients 22 who do not have the PDRC software 194 will be given links to access the software directly over the Internet. In most cases, the PDRC 194 will behave simply as a Netscape NAVIGATOR™ Plug-in or a Microsoft ActiveX™ control or a Java Applet, thus directly integrating the PDRC 194 with the recipient's existing browsers.

Figure 18 illustrates how a portable document send client application and a portable document receive client application are used in the invention. Figure 19 illustrates how a server configuration user interface application is used in the invention.

Portable Document Delivery System Requirements. At the most basic level, a document delivery solution must enable documents to be directed to customers by the producers of these documents, or "pushed". The portable document delivery system 160 is designed so that different types of recipients operating on different computer systems, with different operating systems, E-mail systems, and document types can all benefit from receiving, reading, and using electronic portable documents. The various design parameter categories that the portable document delivery system 160 is adapted for includes primary computer systems (*e.g.* PCs, Workstations, Servers), primary operating systems

(*e.g.* Macintosh, Win 3.1, Win '95, NT, Unix, OS/2), electronic mail systems (*e.g.* Microsoft, cc:Mail, Groupwise, Notes, Eudora), document types (*e.g.* paper, Postscript, Quark, WordPerfect, Excel), and user types (*e.g.* MIS, Legal, Financial, Consumers/Home, MarketingCommunication (MarCom)).

A unique aspect of the portable document delivery system 160 is the level of compatibility the solution provides with all computer systems, operating systems, electronic mail systems, and document types. In one embodiment of the invention, the sender 16 and the receiver 22 of a document are both connected to the Internet. In a preferred embodiment of the invention, the portable document delivery system 160 provides not only an Internet delivery solution, but also backward compatibility with facsimile machines 172 and printers 178, as well as forward compatibility with future distribution print architectures.

Universal Delivery. Delivery solutions must enable users to distribute documents to anyone, which requires support for a variety of computing platforms, compatibility with facsimile 172, and compatibility with future distributed printing architectures. The portable document delivery system 160 can support the conversion and delivery of complex postscript files. Documents can be delivered to any recipient 22 who has an E-mail account and access to the Internet, regardless of the recipient's platform or E-mail system.

Security. Typical applications of document delivery require complete security from the origin of the document complete to the destination. This requirement becomes more pervasive as documents begin to travel across open and wide area networks. The portable document delivery system 160 employs several levels of security. The Portable Document Send Client 192 authenticates and creates a secure socket to upload information to the server 12. Thus, non-BFD servers cannot intercept documents. Additionally, The PDSC 192 allows the

sender 16 to use private and or public encryption to guarantee that only the intended recipients of a document can access those documents. Even in cases where encryption is not used, the portable document delivery system 160 includes sophisticated algorithms to prevent unauthorized users from accessing documents.

Account Management Services. In many instances, document delivery applications cater to businesses where each sender 16 or recipient 22 of a document must be maintained. Consider the case of periodically delivering the documents to the same group of a hundred thousand recipients 22. The sender 16 of the document requires tools to update and manipulate the database of the large subscription/ distribution base.

The portable document delivery system 160 enables publishers 16 to create accounts on BFD servers 12 and then associate transactions with specific accounts 132, 134, 136. The system also enables publishers to consolidate several user accounts into a single billing account 134. Additionally, it allows publishers to associate a specific billing code with transactions which may be consolidated in transaction reports. For example, a law firm could create an account and then a billing code for each client, associating a billing code and account with each document's transaction. The portable document delivery system 160 maintains and updates the account information automatically. A portable document delivery system 160 reporting engine then allows the user to create a report for a given account or for a specific billing code. Such a scheme facilitates client management as well as billing.

Transaction Management Services. Related to account management is the requirement of transaction management. Not only is it necessary to maintain the database of senders 16 and recipients 22 of documents, it is also necessary to provide services to manage the transaction of sending documents.

For example, a sender 16 may want to know if the document was actually delivered and actually received, and perhaps who received the document. In many instances, the publisher 16 would like to charge postage for delivery and will therefore require services to maintain and update accounting information associated with the delivery transactions.

The portable document delivery system 160 is able to create logs associated with each send transaction, and maintain these logs. Each transaction, or document send operation is associated with a specific account. Users 16 can query transaction information directly from the server.

Reporting. Account and transaction management provides no value unless sophisticated means of reporting are provided. For example, users 16 can be provided with a full report of a given transaction, including such information as which documents were delivered to whom, how many users have confirmed delivery of the document, or for billing purposes, the costs associated with the transaction.

Scalability and Bandwidth. Because of the large scope and application of document delivery applications, the portable document delivery system 160 is capable of expanding its capabilities to service millions of documents or recipients 22. Several aspects of the delivery process occur in real time, while other aspects may be deferred or scheduled. In many cases, the portable document delivery system 160 dynamically extends the amount of bandwidth or sets of servers 12a-n deployed to achieve the necessary throughput for document delivery.

The portable document delivery system 160 is scalable to conform with user requirements. The server software is designed to support the sending of millions of documents per day, and is able to exploit whatever bandwidth has

been dedicated to a given server. For example, one current BFD server 12 effectively utilizes 10 Megabit/second bandwidth. The various processes running on BFD servers 12 operate asynchronously, thus allowing for optimal performance on multi-processing servers 12, as well as sophisticated scheduling of the servicing of a given transaction. Special care is taken to operate in real time, particularly for the access of documents from the server 12 by recipients 22.

BFD servers 12 can also distribute work loads across other servers 12a-n. A preferred embodiment of the invention allows individual processes running on a single server 12 to be distributed across a collection of servers 12a-n. In this embodiment, account management processes could run on one server (*e.g.* 12d), while the logging, reporting, transaction management, send, propagate, and retrieve processes run on another server (*e.g.* 12h).

Portable Document Send Client Specification. The Portable Document Send Client (PDSC) 192 allows any computer user to distribute documents directly from the desktop of any personal computer, such as a PC or Macintosh computer. The PDSC 192 integrates directly with all applications 190 through the uses of virtual printer devices, thus enabling the PDSC 192 to be compatible with all applications 192 and formats. Importantly, because the PDSC 192 is integrated directly with portable document technology, the sender 16 of a document need not make assumptions about the capabilities of the intended recipient 22 of the document.

The PDSC 192 allows two primary modes of usage: print or "drag and drop". By print, a sender 16 can simply select the print option from any application 190 and trigger the sequence of events to generate a portable document, and then address and send that document. From the user's perspective, they simply select the print command and are then prompted for the destination of the

document using standard addressing interfaces and address books. A Microsoft Mail™ user, for example, would be prompted with the standard Microsoft Mail™ addressing dialog to direct where a document may be sent. After selecting the destination of the document, the PDSC 192 automatically connects to a BFD server 12 and securely uploads the documents 166 and the intended list of recipients 22, as well as any other attributes selected to customize the send. "Drag and Drop" usage allows users 16 to avoid launching applications and printing to send documents; the document may simply be dropped on a PDSC 192 send icon, which is accessible from the sender's desktop 164.

Additional functionality and customization is one click away. During the addressing process, users 16 are free to customize the options of their send by invoking advanced options. By default, each send will reuse the existing parameters for sending documents. Users 16 can also use the advanced options user interface 193 to customize their delivery options, including, for example, security options and receipt requirements. For example, if the user 16 desires to customize the security options, including private and or public key encryption, the user simply checks a "Public Encrypt" or "Private Encrypt" option. Similarly, the user can select the "Notify on Receipt" option, thus informing the BFD server 12 to confirm delivery when the document is actually received.

BFD Server Configuration Options and User Interface. The BFD Server 12 can be configured and customized directly from a sender desktop 164. The access to the BFD server 12 from the desktop is achieved using an HTML forms user interface. This user interface exists to give server administrators access and control over the advanced options of the BFD server 12. For example, a server administrator might update the database of the 100,000 recipients who are intended to receive a specific document, and then directly invoke the send of the document to those recipients. The server

administrator might generate a report regarding the send transactions which occurred during the previous week.

To access the BFD server 12 from the desktop 166, a user 16 must have a special account created on the BFD server 12, which is created ahead of time by the BFD server 12. Additionally, accessing the BFD server 12 over this account requires several layers of authentication and security, thus preventing unsolicited access.

The Server Configuration User Interface 198 allows the user 16 to access and control the server settings, which may include transaction management, account management, reporting facilities, direct upload and download of documents to distribute, direct manipulation of recipient lists, and direct access to send options.

Portable Document Receive Client. The recipient 22 of a document can utilize the portable document receive client (PDRC) 194 to access and manipulate documents which were sent to the recipient 22 by the portable document send Client 192 or by the BFD server 12 directly via the BFD server administrator. In the event that the recipient 22 of a document does not already have a PDRC 194, the software may be downloaded and installed directly from the Internet. The architecture of the portable document delivery system 160 simplifies this process, and employs dedicated software and scripts, in addition to advents in new browser architectures to enable first-time recipients 22 to be one click away from accessing the necessary software to receive documents.

The most basic case of the portable document receive client 194 can simply function as browser extension, such as a Netscape NAVIGATOR™ plug-in or a Microsoft ActiveX™ control. For other users, the PDRC 194 will behave as a stand alone application which works as a helper application.

A third application exists for portable document delivery system 160 customers who prefer direct access to the portable documents from the recipients desktop 170. In this configuration, a dedicated portable document receive client 194 can be downloaded directly from the Internet. This component will continually monitor the activity of the portable document delivery system 160, and will automatically extract any incoming portable documents from BFD servers 12, and open them for immediate document communication on the computer desktop 170 of the recipient 22.

Recipients 22 of portable documents from the portable document delivery system 160, depending on the send configuration options, will be allowed to view, search, print, archive, or export information from their documents. Documents distributed using Envoy™ or Acrobat™ in conjunction with the portable document delivery system 160 will preserve complete visual fidelity and may be produced on high resolution output devices with the highest level of quality.

Figure 20 illustrates how a document can be sent by the fax gateway 56 of a BFD server 12 to a printer 178. Figure 21 illustrates how a document can be sent by the department gateway 202 of a dedicated corporate BFD server 200 through a LAN 204 to a department printer 178.

Private, Trackable URLs for Directed Document Delivery. This embodiment of the invention provides a unique means of delivering documents electronically. Importantly, this embodiment of the invention enables a number of value added services, in addition to basic document delivery, including but not limited to tracking and security.

The invention provides a document delivery architecture which dynamically generates a private Uniform Resource Locator (URL) to distribute information. Each private URL ("PURL") uniquely identifies the intended recipient of a document, the document or set of documents to be delivered, and (optionally) other parameters specific to the delivery process. The intended recipient of a document uses the PURL to retrieve a document (or documents). The server, upon retrieval of the document, customizes the behavior of the retrieval based upon attributes included in the PURL, as well as log information associated with the retrieval in a data base. This architecture and usage of PURLs enables secure document delivery and tracking of document receipt.

The World Wide Web ("Web") enables consumers to retrieve content from Web servers using Web browsers. In short, consumers pull content from the Web. E-mail enables producers of content to send that content to consumers. In other words, producers push content with e-mail. E-mail Internet servers, as well as the SMTP protocol (simplified mail transport protocol) which governs the behavior of Internet servers, are limited capabilities they provide to users of the Internet. For example, SMTP e-mail servers do not know anything about binary file types, tracking, or security.

The Web and the associated HTTP protocol, by contrast, provides a flexible protocol that enables the efficient, secure transmission of binary information. HTTP, however, is a pull, consumer driven protocol, and hence a producer or sender of information cannot rely on HTTP exclusively to direct the delivery of information.

By combining HTTP for the delivery, as well as using SMTP/e-mail for notification, it is possible to build a solution that allows the producer to be the driver, or to push, but that does not suffer from the limitations and legacy issues associated with SMTP/e-mail.

PURLs are temporary, dynamically generated uniform resource locators which uniquely identify the intended recipient of a document and the document itself, as well attributes associated with the delivery of a document. PURLs avoid attaching information to e-mail messages to send documents, but rather attach a general reference to a document to be sent, and then enable the recipient to access a document via the reference.

When the recipient accesses the document by using the reference, a server can intercept the request to access the document and provide value added services, such as tracking and security. For example, a user can include a key in the PURL that serves to unlock a document on a server, perhaps decrypting an encrypted document. Or, a user can include a unique identification number in the PURL that identifies the recipient. In this case, the server can notice that a specific individual has accessed a specific document, can note that in a data base, and can make that information available to the sender. This embodiment of the invention can therefore provide document tracking.

Figure 22 is a block diagram which depicts a document delivery system that includes private, trackable URLs for directed document delivery according to the invention. A document 310 is forwarded from a sender 300 to a server 315. The server temporarily stores the document. The server dynamically generates a URL for each intended recipient of the document. In addition to encoding user information and document information with the URL, the server also encodes delivery parameters, or transaction identifiers in the URL. Each generated personal URL (PURL) is then forwarded to each intended recipient 320. The recipient is notified 325 that a given document has been sent to him. This typically has the form of an e-mail message which includes a private URL. The recipient, using the PURL 330 and the Web, accesses the document.

When the recipient accesses the document via the PURL, the recipient presents the PURL to the server. The server then has the opportunity determine the next set of actions. For example, the server could notice that the PURL specifies that a password must be presented before the electronic document referenced by the PURL can be accessed. The server may also identify the specific recipient accessing the document by the PURL, and log the fact that the specific recipient has attempted access the specific document, again all identified by the PURL. The server may also log the fact that the entire document was delivered successfully.

Accordingly, a data base maintained on the server has a full log describing the following, for example:

- Who accessed the document;
- When they accessed the document; and
- Whether they successfully accessed the document.

This information which the server has logged can then be reported back to the sender of a document. Hence, using a combination of e-mail for notification, the Web for delivery, and private URLs to identify recipients and documents, a delivery server can be constructed to track documents and report the delivery state of a document back to the sender. The actual implementation of such system may be in accordance with the system herein described in connection with Figures 3-21, or it may take other forms as appropriate.

In other embodiments of the invention, the server can log other types of information. Thus, the server can log the IP address associated with a given recipient who is retrieving a document. The server can also log the IP address

of any subsequent accesses to a given document with the same PURL. Thus, the server could prevent multiple IPs from accessing the same document using the same key. Alternatively, the server could provide a list to the sender containing IP addresses which accessed a specific document intended for a specific recipient.

The above described architecture for delivery also facilitates security. A document can remain encrypted on the server until a recipient presents a valid key to access and decrypt a document. This key is presented as encoded in part of the PURL. Alternatively, the PURL specifies that a key must be retrieved, in which case the server requires that the recipient present a unique password to decrypt the document. In the first case, retrieval of the encrypted document is a one-step, automatic process because the key is encapsulated in the PURL.

PURL Implementation.

First, consider the potential construction of a PURL. The following diagram outlines one specific example of a PURL:

http://posta.tumbleweed.com/cgi/posta.dll? pu = 0-233-33982-FIAAAV4

The above PURL denotes the following:

Value	Meaning
http:/	Use the HTTP protocol to access.
posta.tumbleweed.com	Name of the HTTP server.
cgi/posta.dll	Name of HTTP server extension.
pu=0	Don't use a password.
233	Store item Identifier.
33982	Recipient Identifier.
FIAAAV4	Key to access the document.

With further reference to Figure 22, it should be noted that a PURL 302 is shown having various fields. These fields include a password identifier 331, a store item identifier 332, a recipient identifier 333, a document key 334, and any other optional fields that may be desired 335. These fields are discussed in greater detail below.

Password Identifier. A password identifier specifies whether a password is required to access a given document. In this case, the value "0" indicates no password is required. A value of "1" indicates a password is required.

Store Item Identifier. A store item identifier uniquely identifies which document a given recipient desires to obtain. In this case, the value "233" provides an index into a sparse table on the server, identifying a value which, *e.g.* identifies where a given document resides on the server and/or what a document is named.

Recipient Identifier. A recipient identifier uniquely identifies the intended recipient of a given document. In this case, the value "33982" provides an index into a sparse table on the server. The value at this table index contains recipient information.

Document Key. The document key validates the PURL itself. In this case, the key is a randomly generated number associated with the given recipient and store identifiers. The key is used to validate whether the given recipient identification number is valid, whether the given store identification number is valid, and whether the given recipient with the given store identification number should be granted access to a document. In other embodiments of the invention, the key also encodes an index into a table which contains the validation information, as opposed to encoding the validation information itself.

Importantly, the server has a Web extension, enabling the HTTP processing of a document to be extended to provide customization. Thus, the recipient accessing the document goes through an HTTP server extension to communicate with an HTTP server. This extension, for example, can decide to grant access to a document, in which case it presents the user with a new PURL which facilitates transmission of the specific document.

The server can use the above attributes and values of a PURL to customize the behavior of document delivery. Specifically, the server executes the following steps to deliver the document and record the delivery transaction:

- Decode the PURL into its various parts;
- Validate each component of the PURL;
- Authenticate the PURL using the key;

- Determine which user is accessing the document by using the Recipient Identifier;
- Determine which document the user is accessing by using the Store Item Identifier;
- Determine whether the document, given the above, requires additional input before it can be delivered;
- Deliver the document to the recipient;
- Log all attributes of the transaction, including, *e.g* time of access, success of transmission, and IP of recipient.

Once information has been logged in a data base running on the server which records transaction information, this data can be accessed by the recipient and can even be dynamically transmitted back to the recipient. For example, a given publisher (sender) asks the server's data base for all documents which have been delivered to a specific recipient. The publisher asks the server to generate a report of the status of a given document sent to ten people. The server reports back, for example, that the document has been sent to all ten people at a specific time, but only three of the people have actually retrieved the document. Each document retrieval may include the specific time the document was accessed, the time it was accessed, and whether it was accessed completely and successfully. Hence, dynamically generated PURLs as broadcast over email enable a robust means of tracking the delivery of documents over wide area networks.

Although the electronic document delivery system and its methods of use are described herein in connection with use in the Internet, the invention may be applied to any of a wide variety of networks, including internets, intranets, LANs and WANs, or any combination thereof, as desired. As well, the invention may be applied to a wide variety of computer platforms, communication protocols, portable document formats, or any combination thereof, as desired.

The invention provides a method and system for secure document delivery over a wide area network. A document Delivery Server dynamically retrieves a public key of an intended recipient of a document, then uses the public key to encrypt either a document or the secret key of the document. The server delivers the encrypted document to an intended recipient over a wide area network such as the Internet. The intended recipient decrypts the document using the private key associated with the public key. The invention permits only an intended recipient to gain access to a specific document and therefore provides a unique level of security for document delivery.

For the purposes of the invention, the term document includes any contiguous collection of data, including a stream of data, a video, audio data, an animation, a formatted document such as HTML, PDF, or Envoy, or a data base. While the preferred embodiment of the invention is adapted for use in document transmission over the Internet, the invention is equally applicable to other wide area networks.

Furthermore, while the preferred embodiment of the invention discloses transmission of a document to a recipient computer, the invention is operable for document transmission to any intended recipient maintaining, or having the ability to dynamically generate, a private/public key and to use the private key to decrypt a document encrypted with the corresponding public key. An intended recipient, therefore, includes, for example, an Internet user of a desktop

computer, printer, fax machine, personal digital assistant, or network computer device.

Similarly, while the sender of a document is preferably a desktop computer, the sender also includes any device capable of encrypting a document and communicating with the Delivery Server, such as a network computer device. In an alternative embodiment of the invention, the document is encrypted by the delivery server. In this embodiment, the sender includes any device, such as an Internet browser device, Internet telephone device, personal digital assistant, or fax machine, that can transmit a document to the Delivery Server for encryption and transmission to the intended recipient.

Figure 23 is a diagram of a system for dynamic server document encryption, according to a first preferred embodiment of the invention. A document stored on a desktop computer, the sender 1032, is to be transmitted to another computer, the intended recipient 1034. In this first preferred embodiment, the document is stored in Portable Document Format (PDF). However, in alternative embodiments, a document may be stored in any appropriate format. Portable Document (PD) formats are required for distributed print and fax solutions. However, PD formats are not required for the invention.

The document is sent from sender to recipient via the Delivery Server 1036. In this first preferred embodiment of the invention, the Delivery Server is directed by the sender to communicate with a certificate authority database server 1038 to retrieve the intended recipient's public key (certificate). The Delivery Server dynamically queries the certificate authority and retrieves the public key. The public key is transmitted to the Delivery Server and from there to the sender. In alternative embodiments of the invention, the Delivery Server retrieves the intended recipient's public key from the intended recipient's desktop computer,

an Internet server, or from an intranet server connected to the intended recipient's desktop computer.

In the first preferred embodiment of the invention, the sender encrypts the document using a secret key and uses the public key to encrypt the secret key. The document and encrypted secret key are then transmitted to the intended recipient. The secret key is decrypted with the intended recipient's private key and is then used to decrypt the document.

In an alternative, equally preferred embodiment, the sender uses the public key to encrypt the document. The encrypted document is then transmitted to the intended recipient and decrypted using the private key associated with the public key.

Figure 24 is a flow chart of the set of operations for dynamic server document encryption, according to a first preferred embodiment of the invention. In the example, the sender encrypts the document 1040 using a secret key. Such secret key includes any appropriate encryption scheme known in the prior art. The sender then contacts a Delivery Server 1045 to query 1050 the public key associated with the intended recipient. The Delivery Server retrieves this certificate in real time 1055, for example from the data base of a certificate authority, and transmits the certificate back to the sender 1060.

In the event that the certificate authority returns no certificate, the Delivery Server dynamically generates a new certificate for the recipient. To do so, the Delivery Server forwards a dynamically generated URL in an e-mail message to the recipient. Recipient access of the URL dynamically retrieves a Java Applet or Plug-in, which is automatically downloaded to the recipient's system. This applet or Plug-in then runs on the Recipient system and constructs a

private/public key pair. Generating a private/public key pair on a local machine is not specific to this invention and is documented in a number of sources.

The applet or plug-in next forwards the public key to the Delivery Server. The server, using properties of the generated URL, identifies the e-mail address of the recipient. Thus, the generated public key has the property of having authenticated the e-mail address of the recipient, as the URL to invoke the key generation has only been forwarded to a specific e-mail address. The server combines the e-mail address and public key into a certificate and returned to the Send Client or used by the server to encrypt the document or secret key. The Delivery Server, using LDAP or a similar protocol, may communicate the certificate to the certificate authority. Alternatively, the Delivery Server simply may maintain a local database or dynamically generated certificates for future use.

Upon receiving the public key from the Delivery Server, the sender encrypts the secret key 65 with the public key. In an alternative, equally preferred embodiment of the invention, the sender does not encrypt the document until the public key has been received. Because the document is not encrypted if the public key is not authenticated, this embodiment minimizes processing time when a public key cannot be retrieved.

The sender then forwards 1070 the encrypted document, the address of the intended recipient (for example an email address), delivery instructions, and the encrypted secret key to the Delivery Server over a secure channel. Thus, the document does not leave the Sender until the document has been encrypted with the secret key and the secret key has been encrypted with the intended recipient's public key. The Delivery Server then delivers 1075 the encrypted document and secret key to the intended recipient. The intended recipient, using the private key associated with the public key, decrypts the secret key

1080 and uses the secret key to decrypt the document. Such scheme prevents unauthorized access to the document, since the document can only be accessed by the owner of the public key.

Figure 25 is a flow chart of the set of operations for dynamic server document encryption, according to an alternative embodiment of the invention. The sender notifies the Delivery Server 1090 that the sender intends to send a document to a given recipient. The Delivery Server queries 1095 the certificate authority to obtain the intended recipient's public key, which is returned 1100 to the Delivery Server.

In this embodiment, the Sender does not encrypt the document but forwards the document 1105 to the Delivery Server over a secure channel. The Delivery Server then encrypts 1110 the document using a secret key. The Delivery Server uses the retrieved public key of the intended recipient to encrypt 1115 the secret key, and then forwards the encrypted document and secret key to the intended recipient 1120. The intended recipient uses the private key to decrypt the secret, and then uses the secret key to decrypt the document 1125.

Alternatively, the Delivery Server may use the public key to encrypt the document. The encrypted document is then transmitted to the recipient.

In the preferred implementation of the invention, the sender is connected to the intended recipient via a Delivery Server, all running over a wide area network, such as the Internet. The sender is preferably a computer using software referred to herein as the Send Client.

The Delivery Server is responsible for determining the public key of a given recipient and forwarding that key to the Send Client. The Delivery Server is

also responsible for delivering the encrypted document and secret key to the intended recipient.

The Send Client initiates the delivery transaction by first identifying the document to be delivered, any delivery parameters, and the set of intended recipients to receive the document. Delivery parameters include such options as the scheduled delivery time, security options, urgency of the delivery, presentation parameters for the delivery, and receipt notification.

The Send Client then initiates a dialog with the Delivery Server and encrypts the document with a secret key. The dialog and encryption steps may be performed simultaneously or sequentially, depending upon the sender's hardware and software configuration. In the dialog, the Send Client forwards to the Delivery Server the intended recipient(s) of a given document. The Send Client requests that the Delivery Server contact the Send Client once the public key has been acquired.

The Send Client expresses the identity of the intended recipient(s) of a given document in different ways. In the preferred embodiment of the invention, the Send Client uses the electronic mail (email) address of the intended recipient as the identifier of the intended recipient. However, the Send Client can also identify the intended recipient with an alternative identifier, such as a driver's license number, a social security number, an abstract identifier, a symbol name, or a fax number.

The Delivery Server uses several techniques to obtain the certificate for the intended recipient. In the preferred embodiment of the invention, the Delivery Server contacts a certificate authority data base server, presents information identifying the intended recipient, and asks for the intended recipient's public key. The invention may therefore be used to obtain information from certificate

authorities that maintain public key data bases that can be accessed dynamically over a programmatic interface (queried) in real time.

The invention is implemented using any appropriate means for a Delivery Server to query a public key of an intended recipient in real time without user intervention. Thus, the specific protocol and means of accessing the public key data base are not significant for the invention. The public key data base is preferably accessed using the Internet Lightweight Directory Access Protocol (LDAP) standard developed by the University of Michigan in conjunction with the Internet Engineering Task Force. LDAP servers provide directory and other services. Using LDAP protocol, a given server may be queried, and information maintained on that server may be retrieved over an electronic network. LDAP servers can be queried directly using standard Internet protocols. Alternative embodiments of the invention use, for example, SQL Queries with different connectivity protocols including RPC (remote procedure call).

The certificate authority data base server and the Delivery Server may be either the same or separate servers. Maintaining both the certificate authority data base and the Delivery Server on the same server is advantageous for a dedicated application of document delivery which does not require access to a general data base of certificates. For example, a corporation may maintain a database of employees' public keys on the same server used for Internet communications. The same server is therefore used as the certificate authority data base and as the Delivery Server for interoffice communication within the company.

For embodiments in which the certificate authority data base server and the Delivery Server are separate, the Delivery Server may maintain a cache or local copy of recently queried certificates. Use of such cache saves time in future queries for the same recipient and certificate.

The invention supports document delivery to one or more recipients. For multiple recipients, the process discussed above is applied in batch mode. An ordered list of intended recipients is forwarded to the Delivery Server, and the Delivery Server returns a corresponding ordered list of certificates.

The invention may also be used to send multiple documents from sender to recipient(s). In such case, a single secret key is used to encrypt each document. Once the Delivery Server has returned a certificate containing each recipient's public key, the single secret key is encrypted with the retrieved public key(s) of the intended recipient(s). For each recipient, the Send Client forwards an encrypted secret key and the encrypted document(s) to the Delivery Server, along with the intended recipient address and delivery parameters.

The Delivery Server then forwards to each recipient the combined encrypted secret key and document(s). The recipient device uses software known as the Receive Client. The Receive Client is currently implemented as a Java Applet as well as a plug-in to standard internet browsers. Java is a programming language developed by Sun Microsystems of Mountain View, CA. However, the Receive Client may also be implemented using any other programming language that is capable of receiving and decrypting the transmitted secret key and document(s).

When implemented as a Java Applet, the Receive Client is distributed dynamically from the Delivery Server to the intended recipient's system. The Receive Client uses the private key to decrypt the secret key. This decrypted secret key is then used by the recipient to decrypt the document(s).

In the preferred embodiment of this invention, the Receive Client accesses the encrypted secret key and document from the Delivery Server using Hypertext

Transmission Protocol (HTTP), the standard internet delivery protocol. However, the Receive Client may access the Delivery Server using any other appropriate protocol.

When using HTTP, the Receive Client is sent a uniform resource locator (URL) containing the address of the documents and key to be delivered. In the preferred embodiment of the invention, the document(s) and secret key are packaged into a single file or stream of data, which is delivered intact to the Receive Client using HTTP. The Receive Client is thereby given maximal flexibility to retrieve the package and decrypt it from the recipient(s) web browser. The recipient may use any web browser or other software application that is capable of receiving the data transmitted over the wide area network.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the scope of the present invention.

The source code for the Send Client, the Receive Client, and for the Delivery Server software can be readily configured by one skilled in the art using well-known programming techniques and hardware components. Additionally, Send Client and Delivery Server functions may also be accomplished by other means, including integrated circuits and programmable memory devices such as an EEPROM.

The implementation of the dynamic server document encryption discussed above with regard to the preferred embodiment of the invention is only one possible implementation. Alternate embodiments may use other implementations consistent with the teachings of the invention.

The Receive Client may be configured to direct a document to another device. For example, a decrypted document may be sent to a printer or a fax machine.

The invention may use any appropriate encryption scheme for the secret key, public key, and private key, including the RSA and Verisign schemes.

Although the present invention has been described in detail with reference to a particular preferred embodiment, persons possessing ordinary skill in the art to which this invention pertains will appreciate that various modifications and enhancements may be made without departing from the spirit and scope of the claims that follow.

CLAIMS

1. An apparatus for delivering an electronic document between a sending computer and a receiving computer, comprising:

a server interposed between said sending computer and said receiving computer, wherein when said electronic document is forwarded to said server from said sending computer and said server dynamically generates a private Uniform Resource Locator ("PURL") to distribute said electronic document.

2. The apparatus of Claim 1, wherein said PURL uniquely identifies an intended recipient of said electronic document and, optionally, other parameters specific to said electronic document delivery.

3. The apparatus of Claim 2, wherein said intended recipient of said electronic document uses said PURL to retrieve said electronic document.

4. The apparatus of Claim 3, wherein said server, upon retrieval of said electronic document, customizes the behavior of said retrieval based upon attributes included in said PURL and, optionally, log information associated with said retrieval in a data base, to enable secure document delivery and tracking of document receipt.

5. The apparatus of Claim 1, wherein said server uses HTTP for delivery of said electronic document and SMTP/e-mail for notification of arrival at said server of said electronic document.

6. The apparatus of Claim 1, said PURL comprising:

a temporary, dynamically generated uniform resource locator which uniquely identifies an intended recipient of said electronic document and,

optionally, said electronic document itself and attributes associated with delivery of said electronic document.

7. The apparatus of Claim 1, wherein said PURL attaches a general reference to an electronic document to be sent, and enables a recipient to access said electronic document via said reference.

8. The apparatus of Claim 7, wherein said server intercepts said request to access said electronic document and provides a value added service in connection with said access, when said recipient accesses said document by using said reference.

9. The apparatus of Claim 1, said PURL further comprising:
a key that unlocks a document on said server.

10. The apparatus of Claim 1, said PURL further comprising:
a unique identification number that identifies a recipient of said electronic document.

11. The apparatus of Claim 10, wherein said server notices that a specific individual has accessed a specific document and notes that in a data base to provide document tracking.

12. A document delivery system for delivering an electronic document between a sender and at least one recipient, comprising:
a document server that temporarily stores said electronic document, wherein said server dynamically generates a private, trackable URL ("PURL") for each intended recipient of said document that is forwarded to each intended recipient.

13. The system of Claim 12, wherein said server encodes delivery parameters, or transaction identifiers in said PURL.

14. The system of Claim 12, wherein said PURL comprises:
an e-mail message.

15. The system of Claim 12, wherein said recipient accesses said electronic document via said PURL by presenting said PURL to said server.

16. The system of Claim 15, wherein said server requires that said PURL specifies that a password must be presented before the electronic document referenced by said PURL can be accessed.

17. The system of Claim 15, wherein said server identifies a specific recipient accessing said electronic document with said PURL.

18. The system of Claim 17, wherein said server logs the fact that a specific recipient has attempted access a specific document.

19. The system of Claim 17, wherein said server logs the fact that an entire electronic document was delivered successfully.

20. The system of Claim 12, further comprising:
a data base maintained on, or associated with, said server that has a full log describing any of who accessed said electronic document, when they accessed the document, and whether they successfully accessed said electronic document.

21. The system of Claim 20, wherein information which said server has logged is reported back to the sender of an electronic document.

22. The system of Claim 12, wherein said server can log any of the IP address associated with a given recipient who is retrieving a document, the IP address of any subsequent accesses to a given document with said same PURL, or a list containing IP addresses which accessed a specific document intended for a specific recipient.

23. The system of Claim 12, wherein said electronic document remains encrypted on said server until a recipient presents a valid key to access and decrypt said electronic document, wherein said key is presented as encoded in part of said PURL.

24. The system of Claim 12, wherein said PURL specifies that a key must be retrieved, and wherein said server requires that a recipient present a unique password to decrypt said electronic document.

25. The system of Claim 12, said PURL further comprising:

a password identifier that specifies whether a password is required to access a given document.

26. The system of Claim 12, said PURL further comprising:

a store item identifier that uniquely identifies which document a given recipient desires to obtain.

27. The system of Claim 12, said PURL further comprising:

a recipient identifier that uniquely identifies an intended recipient of a given document.

28. The system of Claim 12, said PURL further comprising:

a document key that validates said PURL itself.

29. The system of Claim 28, said document key further comprising:

a key that is a randomly generated number associated with a given recipient and a document a given recipient desires to obtain, wherein said key is used to validate whether a given recipient identification number is valid, whether a given store identification number is valid, and whether a given recipient with a given store identification number should be granted access to a document.

30. A method for delivering an electronic document between a sender and at least one recipient, comprising the steps of:

using a document server to temporarily store said electronic document, wherein said server dynamically generates a private, trackable URL ("PURL") for each intended recipient of said document that is forwarded to each intended recipient;

decoding said PURL into its component parts; and

validating each component part of said PURL.

31. The method of Claim 30, further comprising the step of:

authenticating PURL using a key.

32. The method of Claim 31, further comprising the step of:

determining which user is accessing a document by using a recipient identifier that uniquely identifies an intended recipient of a given document.

33. The method of Claim 32, further comprising the step of:

determining which document a user is accessing by using a store item identifier that uniquely identifies which document a given recipient desires to obtain.

34. The method of Claim 33, further comprising the step of:

determining whether said document requires additional input before it can be delivered.

35. The method of Claim 35, further comprising the step of:

delivering said document to said recipient.

36. The method of Claim 31, further comprising the step of:

logging all attributes of a delivery transaction, including any of time of access, success of transmission, and IP of recipient.

37. A method for secure document delivery from a sender over a wide area network, comprising the steps of:

a sender encrypting a document using a secret key;

the sender contacting a Delivery Server to query a public key associated with an intended recipient;

the Delivery Server dynamically retrieving the public key in real time;

the Delivery Server transmitting the public key back to the sender;

the sender encrypting the secret key with the public key; and

the sender transmitting the encrypted document and the encrypted secret key to the Delivery Server for transmission to the recipient.

38. The method of Claim 37, further comprising the step of the recipient decrypting the secret key using a private key.

39. The method of Claim 38, further comprising the step of the recipient decrypting the document using the secret key.

40. The method of Claim 37, wherein the sender encrypts the document prior to receiving the public key from the Delivery Server.

41. The method of Claim 37, wherein the sender encrypts the document subsequent to receiving the public key from the Delivery Server.

42. The method of Claim 37, wherein the wherein the document is one of a contiguous collection of data, a stream of data, a video, audio data, an animation, a formatted document, or a data bas.

43. The method of Claim 37, further comprising the step of the sender forwarding the address of the intended recipient and document delivery instructions to the Delivery Server.

44. The method of Claim 37, wherein the wide area network is the Internet.

45. The method of Claim 37, wherein the recipient is one of a desktop computer, a printer, a fax machine, a personal digital assistant, or a network computer device.

46. The method of Claim 37, wherein the sender is one of a desktop computer, an Internet browser device, an Internet telephone device, or a network computer device.

47. The method of Claim 37, wherein the database server dynamically retrieves the public key from one of a certificate authority, an Internet server,

personal digital assistant, the intended recipient's desktop computer, or from an intranet server connected to the intended recipient's desktop computer.

48. A method for secure document delivery from a sender over a wide area network, comprising the steps of:

a sender contacting a Delivery Server to query a public key associated with an intended recipient of a document;

the Delivery Server dynamically retrieving the public key in real time;

the Delivery Server transmitting the public key back to the sender;

the sender encrypting the document with the public key; and

the sender transmitting the encrypted document to the Delivery Server for transmission to the recipient.

49. The method of Claim 48, further comprising the step of the recipient decrypting the document using a private key.

50. The method of Claim 48, wherein the recipient is one of a desktop computer, a printer, a fax machine, a personal digital assistant, or a network computer device.

51. The method of Claim 48, wherein the sender is one of a desktop computer, an Internet browser device, an Internet telephone device, or a network computer device.

52. The method of Claim 48, wherein the database server dynamically retrieves the public key from one of a certificate authority, an Internet server, personal digital assistant, the intended recipient's desktop computer, or from an intranet server connected to the intended recipient's desktop computer.

53. A method for secure document delivery from a sender over a wide area network, comprising the steps of:

a sender contacting a Delivery Server to query a public key associated with an intended recipient;

the Delivery Server dynamically retrieving the public key in real time;

the sender transmitting the document to the Delivery Server ;

the Delivery Server encrypting the document with a secret key and encrypting the secret key with the public key; and

the Delivery Server transmitting the encrypted secret key and the encrypted document to the intended recipient.

54. The method of Claim 53, further comprising the step of the recipient decrypting the secret key using a private key.

55. The method of Claim 54, further comprising the step of the recipient decrypting the document using the secret key.

56. The method of Claim 53, wherein the recipient is one of a desktop computer, a printer, a fax machine, a personal digital assistant, or a network computer device.

57. The method of Claim 53, wherein the sender is one of a desktop computer, a network computer device, an Internet browser device, an Internet telephone device, or a fax machine.

58. The method of Claim 53, wherein the database server dynamically retrieves the public key from one of a certificate authority, an Internet server, personal digital assistant, the intended recipient's desktop computer, or from an intranet server connected to the intended recipient's desktop computer.

59. The method of Claim 58, further comprising the step of:

dynamically generating a public key at said Delivery Server where said recipient does not have a public key at the time of said retrieval.

60. The method of Claim 59, said dynamic generating step further comprising the steps of:

forwarding a message to said recipient, the reading of which retrieves a module that constructs a private/public key pair on said recipient's system.

61. The method of Claim 60, said dynamic generating step further comprising the step of:

forwarding said public key from said recipient's system to said Delivery Server.

62. A system for secure document delivery from a sender over a wide area network, comprising:

a Delivery Server for querying a public key associated with an intended recipient at the direction of a sender, the Delivery Server dynamically retrieving the public key in real time and transmitting the public key back to the sender;

the sender for encrypting a document using a secret key, the sender encrypting the secret key with the public key and the sender transmitting the encrypted document and the encrypted secret key to the Delivery Server for transmission to the intended recipient.

63. The system of Claim 62, further comprising:

means for decrypting the secret key by the recipient using a private key; and

means for decrypting the encrypted document using the secret key.

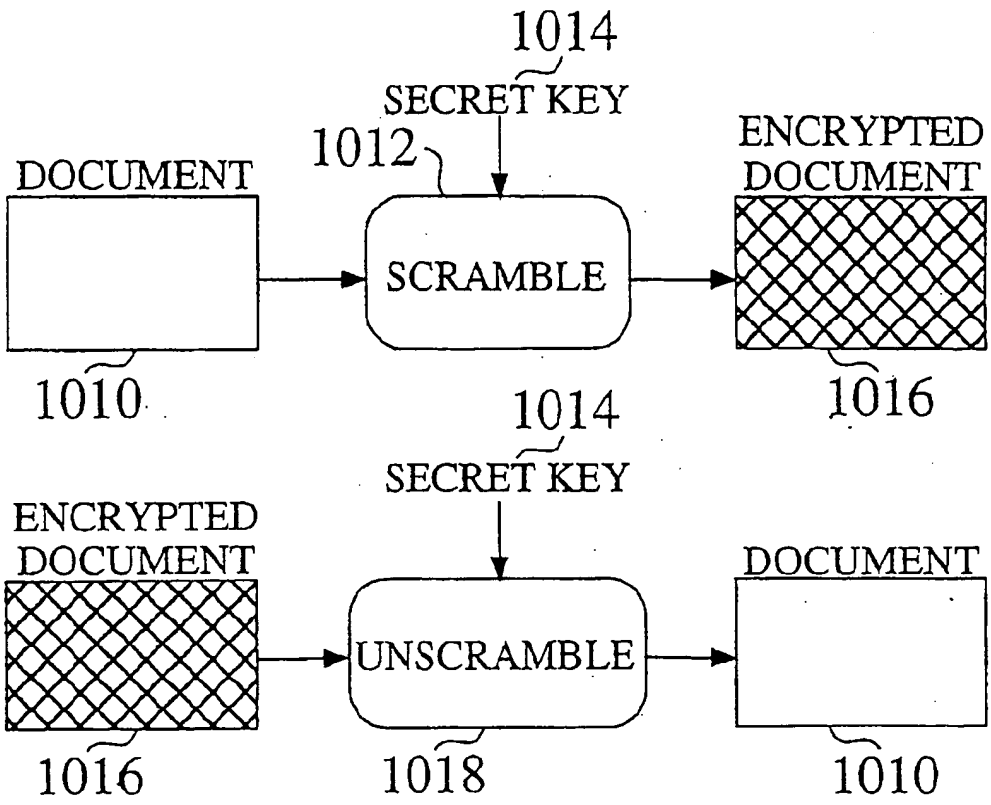


Fig. 1
(PRIOR ART)

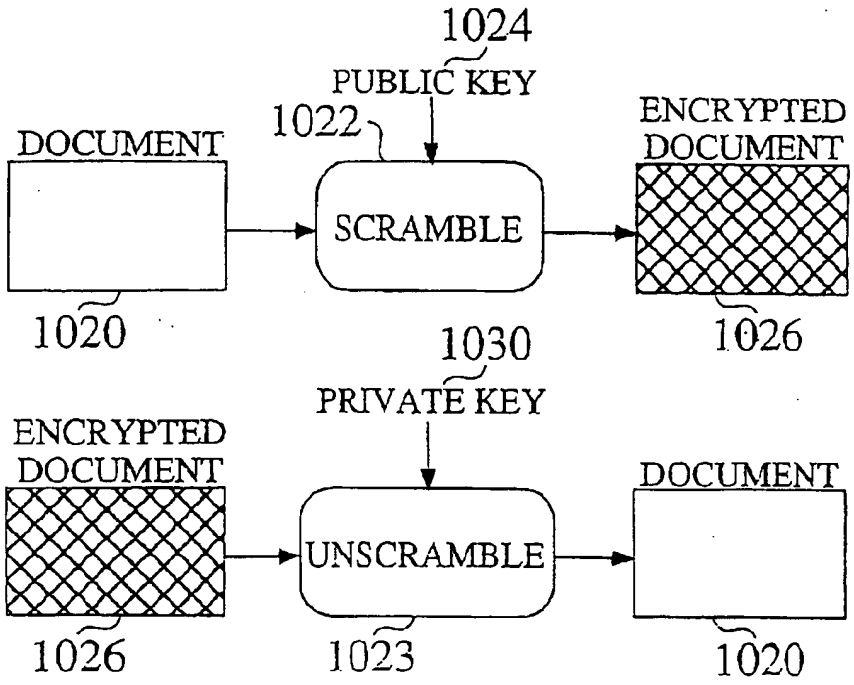

1024
PUBLIC KEY
1022
DOCUMENT
SCRAMBLE
ENCRYPTED DOCUMENT
1020
1026
1030
PRIVATE KEY
ENCRYPTED DOCUMENT
UNSCRAMBLE
DOCUMENT
1026
1023
1020

Fig. 2

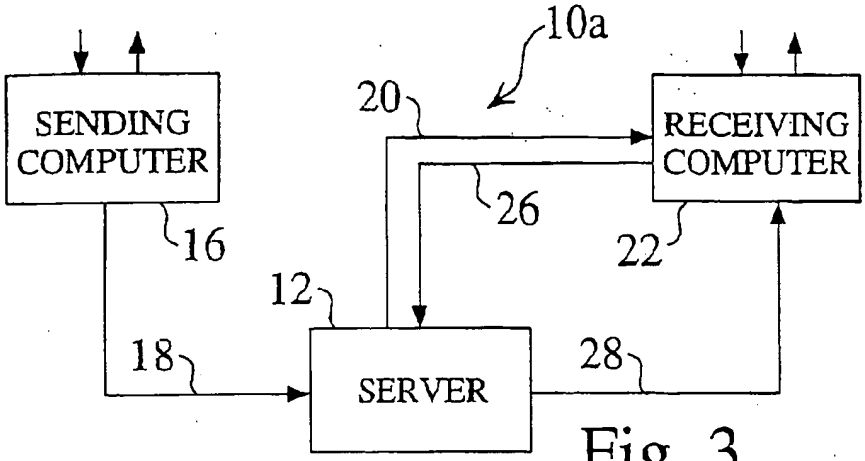


Fig. 3

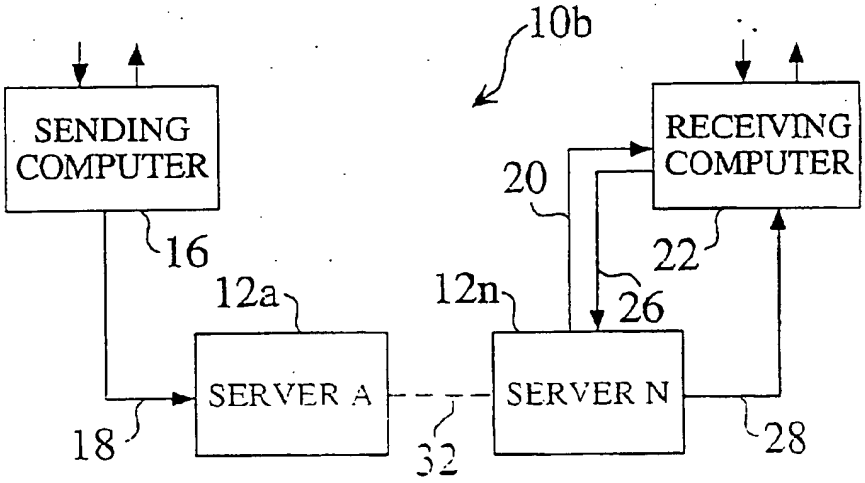


Fig. 4

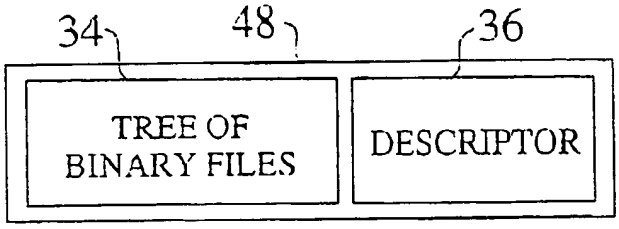


Fig. 5

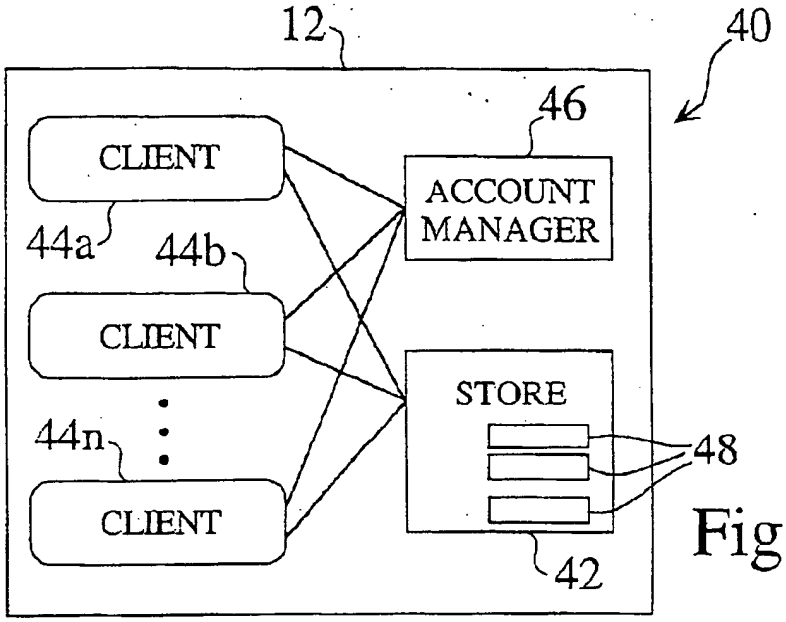

12
40
46
CLIENT
ACCOUNT MANAGER
44a
44b
CLIENT
STORE
44n
48
CLIENT
42

Fig. 6

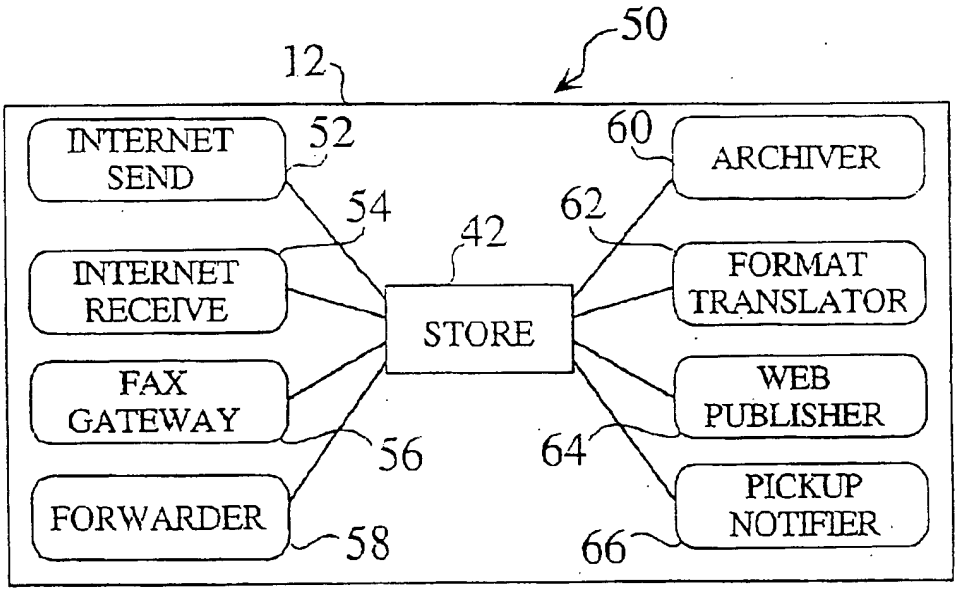

50
12
INTERNET SEND
52
60
ARCHIVER
54
62
42
INTERNET RECEIVE
STORE
FORMAT TRANSLATOR
FAX GATEWAY
WEB PUBLISHER
56
64
FORWARDER
PICKUP NOTIFIER
58
66

Fig. 7

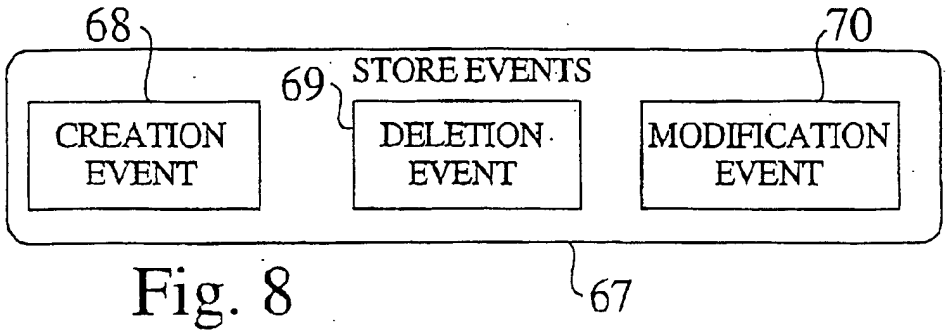

68
70
69
STORE EVENTS
CREATION EVENT
DELETION EVENT
MODIFICATION EVENT
67

Fig. 8

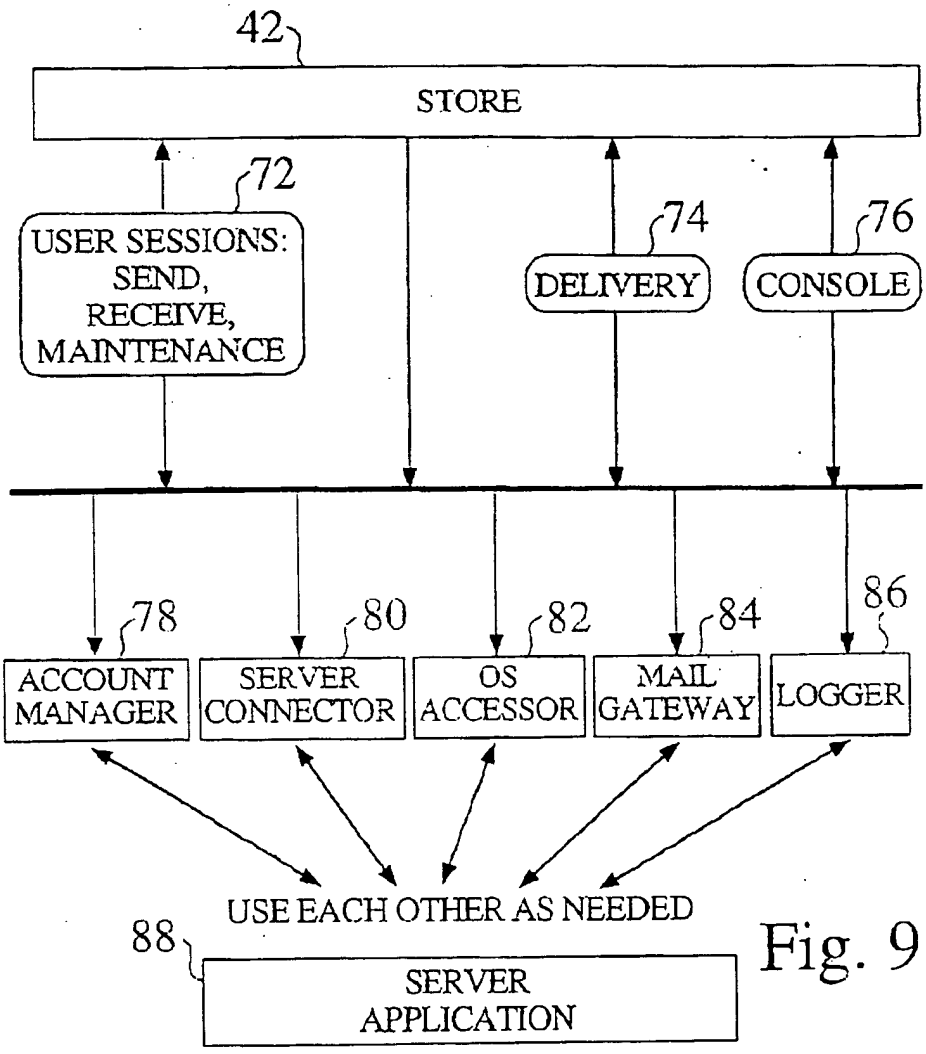

42
STORE
72
USER SESSIONS: SEND, RECEIVE, MAINTENANCE
74
DELIVERY
76
CONSOLE
78
ACCOUNT MANAGER
80
SERVER CONNECTOR
82
OS ACCESSOR
84
MAIL GATEWAY
86
LOGGER
USE EACH OTHER AS NEEDED
88
SERVER APPLICATION

Fig. 9

6

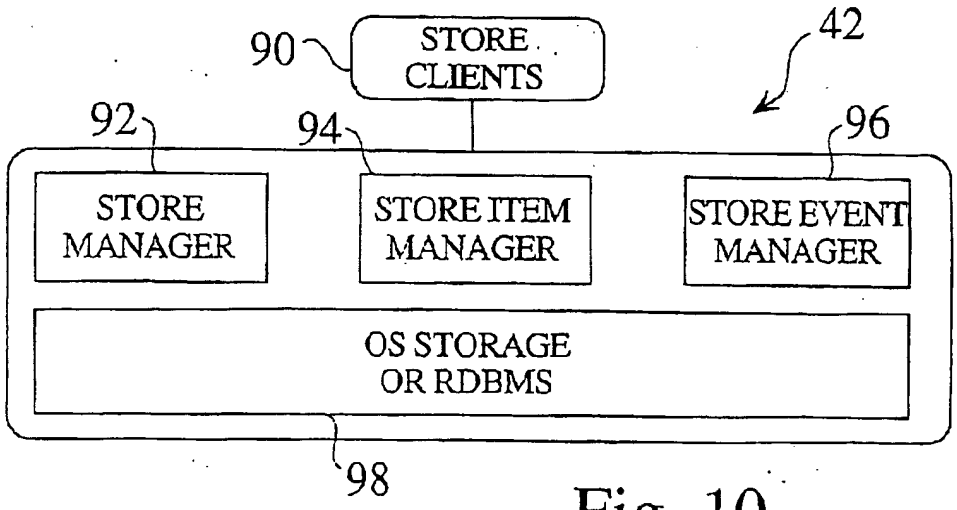


Fig. 10

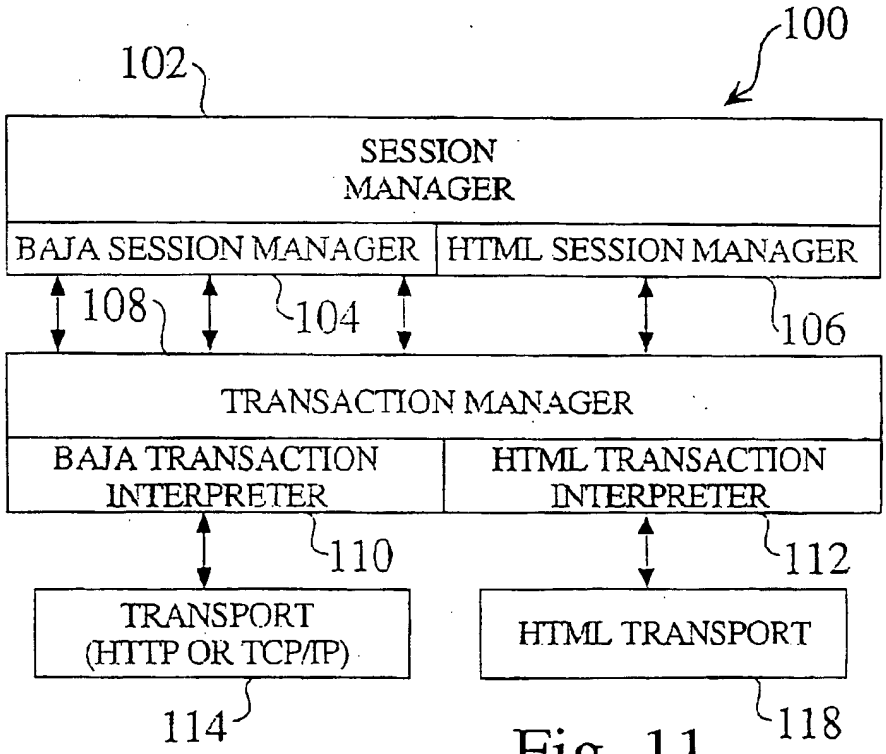


Fig. 11

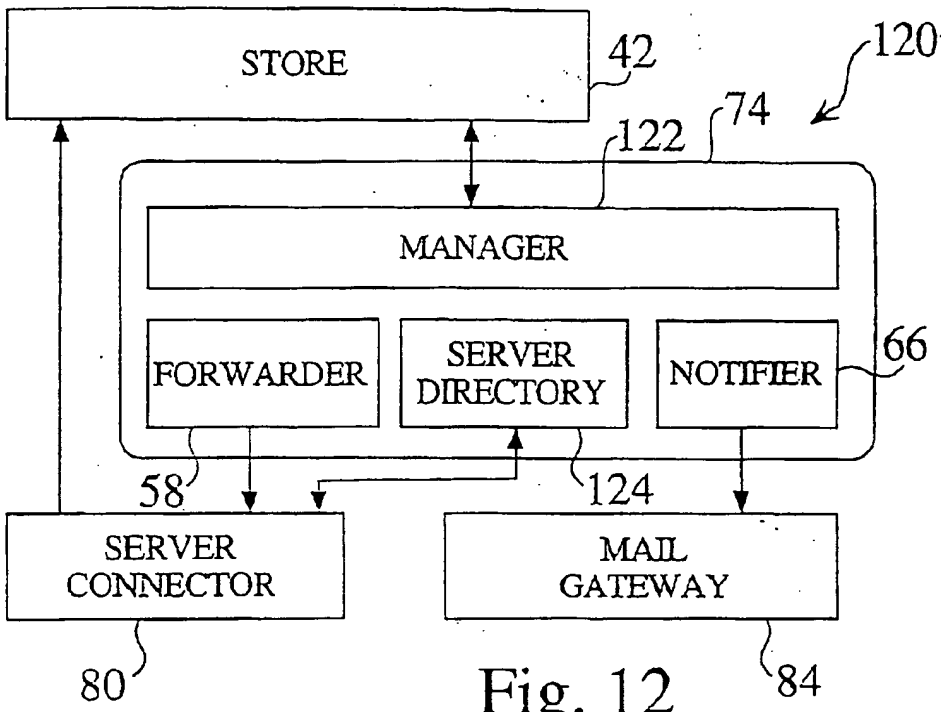

STORE
42
120
74
122
MANAGER
FORWARDER
SERVER DIRECTORY
NOTIFIER
66
58
124
SERVER CONNECTOR
MAIL GATEWAY
80
84

Fig. 12

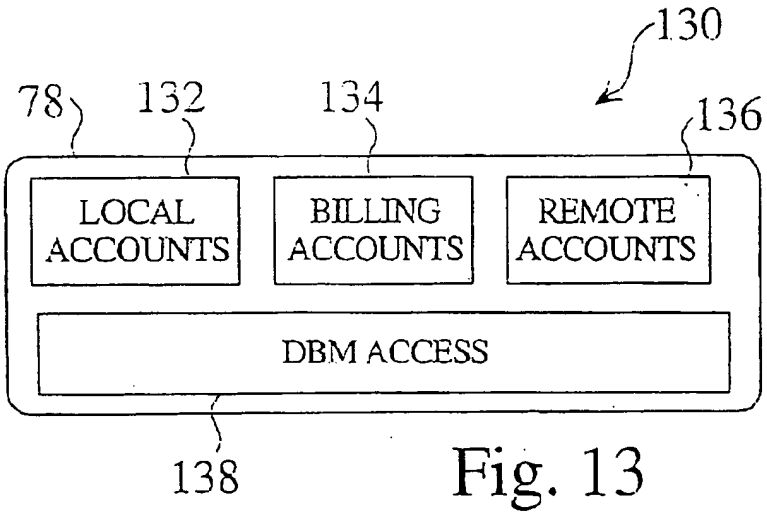

130
78
132
134
136
LOCAL ACCOUNTS
BILLING ACCOUNTS
REMOTE ACCOUNTS
DBM ACCESS
138

Fig. 13

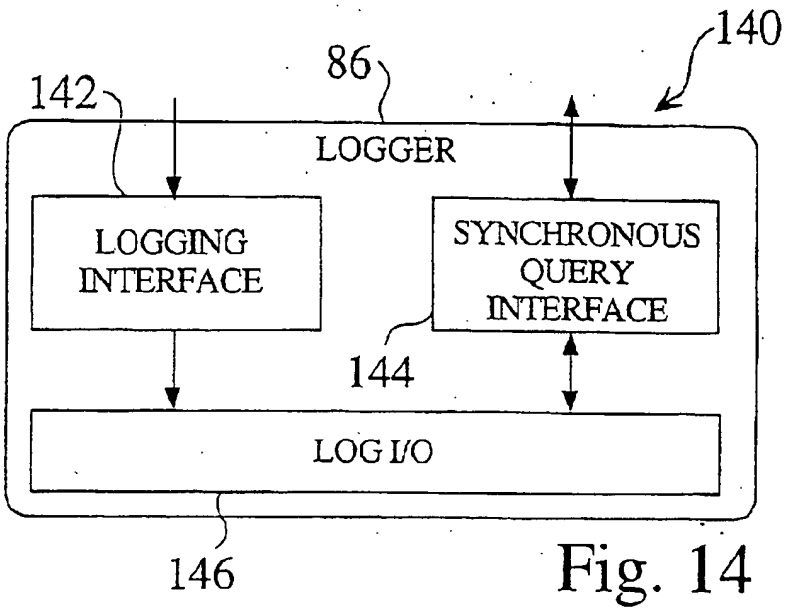

140
86
142
LOGGER
LOGGING INTERFACE
SYNCHRONOUS QUERY INTERFACE
144
LOG I/O
146

Fig. 14

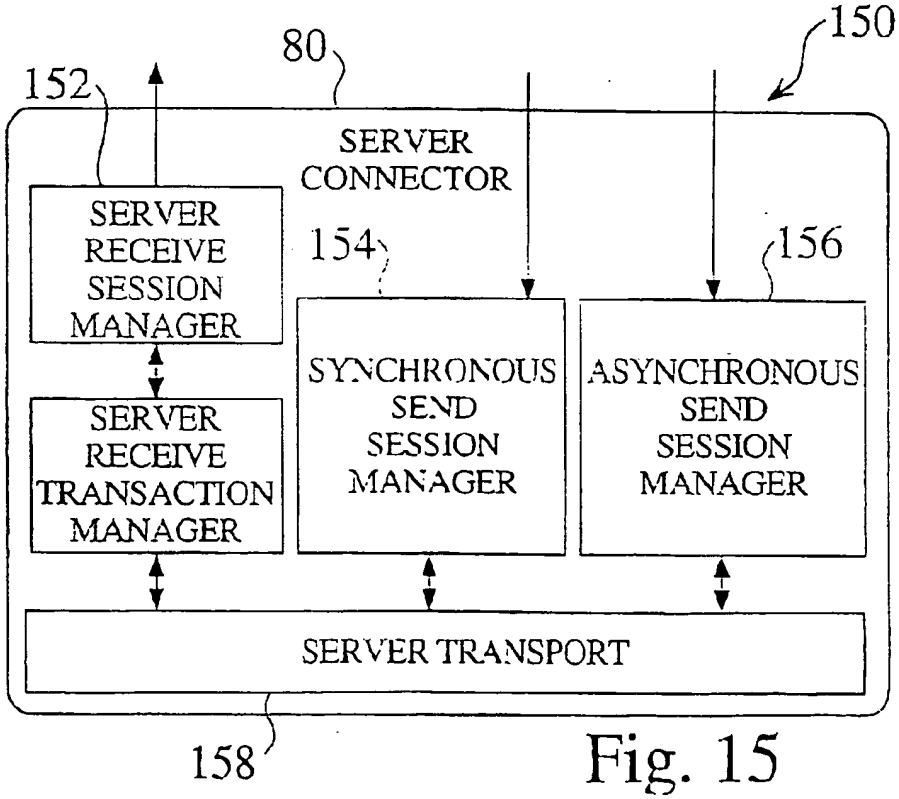

150
80
152
SERVER CONNECTOR
SERVER RECEIVE SESSION MANAGER
154
156
SYNCHRONOUS SEND SESSION MANAGER
ASYNCHRONOUS SEND SESSION MANAGER
SERVER RECEIVE TRANSACTION MANAGER
SERVER TRANSPORT
158

Fig. 15

9

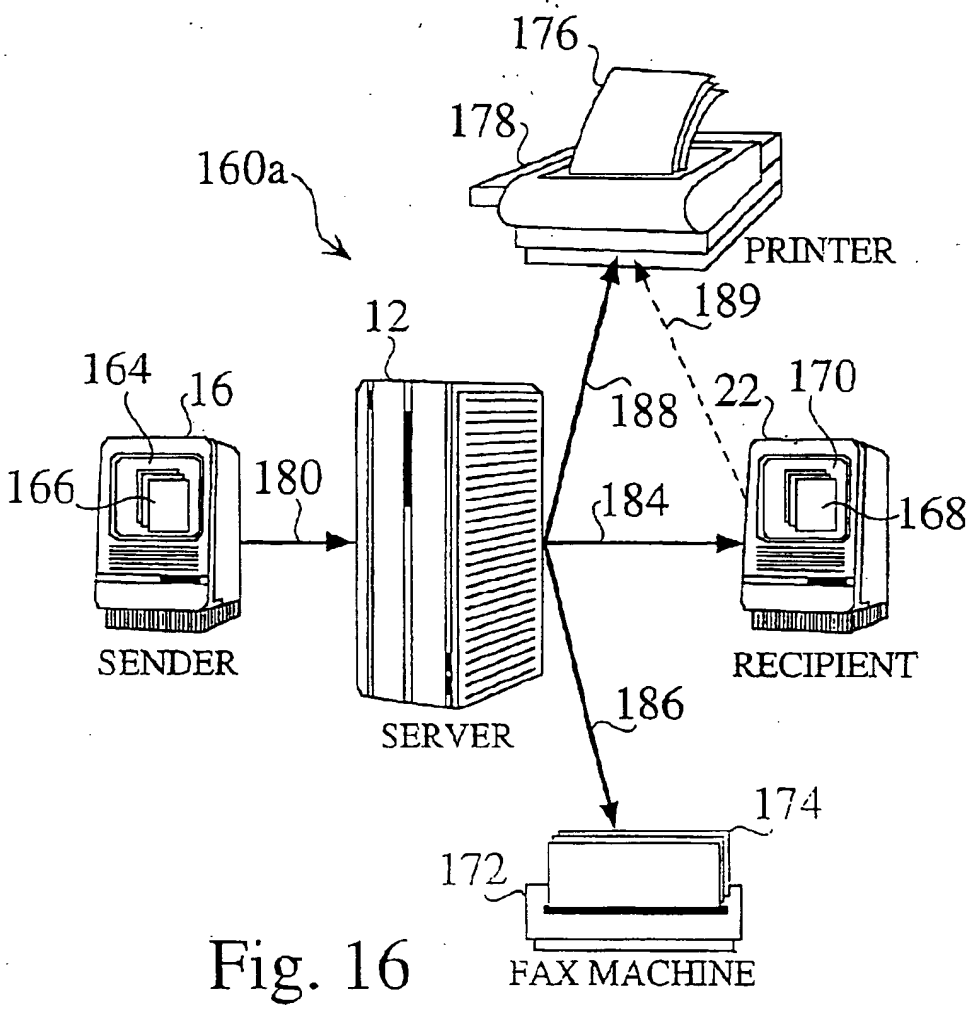

176
178
160a
PRINTER
189
12
164
16
188
22
170
166
180
184
168
SENDER
SERVER
RECIPIENT
186
174
172
FAX MACHINE

Fig. 16

10

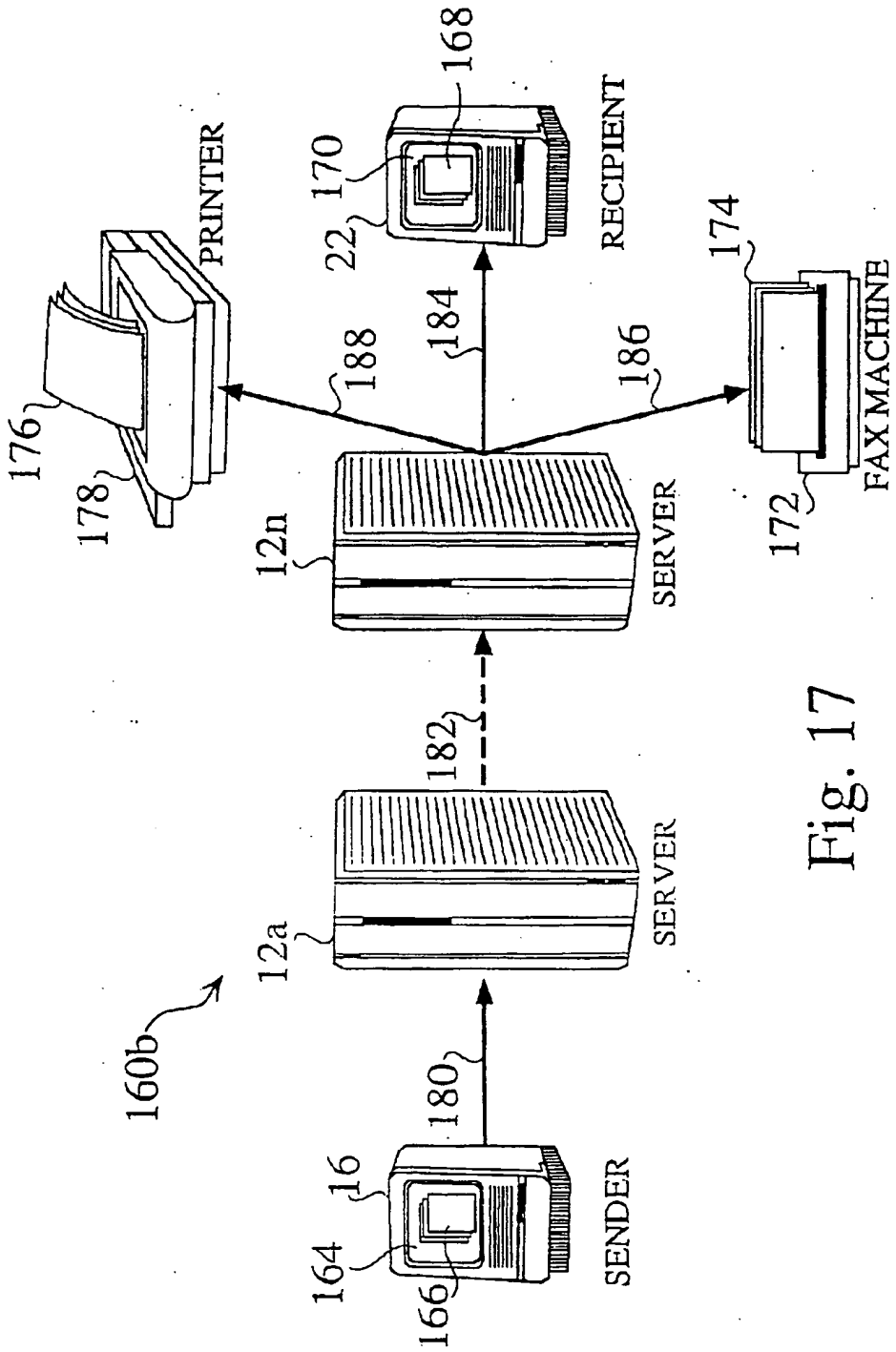

160b
SENDER
16
164
166
180
SERVER
12a
182
SERVER
12n
PRINTER
176
178
188
RECIPIENT
168
170
22
184
FAX MACHINE
174
172
186

Fig. 17

11

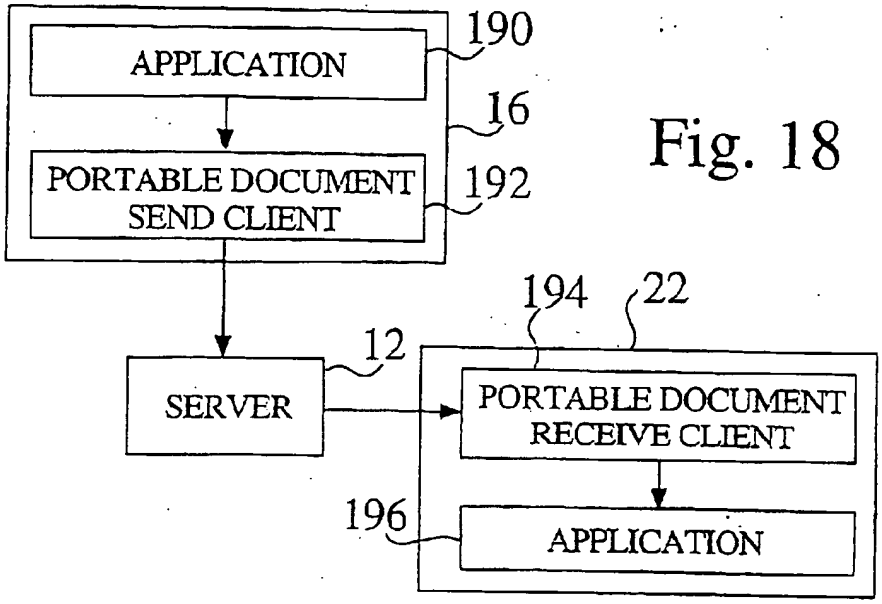


Fig. 18

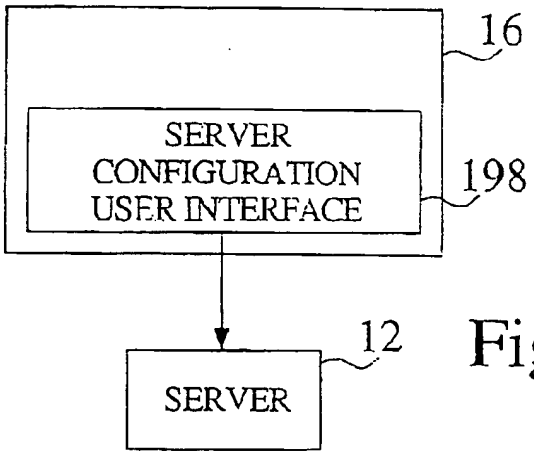


Fig. 19

12

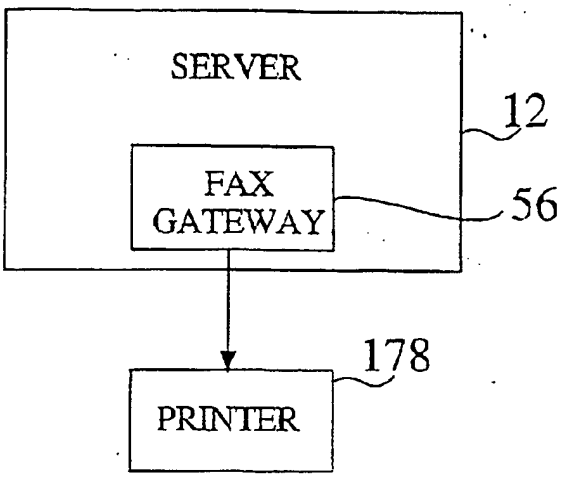


Fig. 20

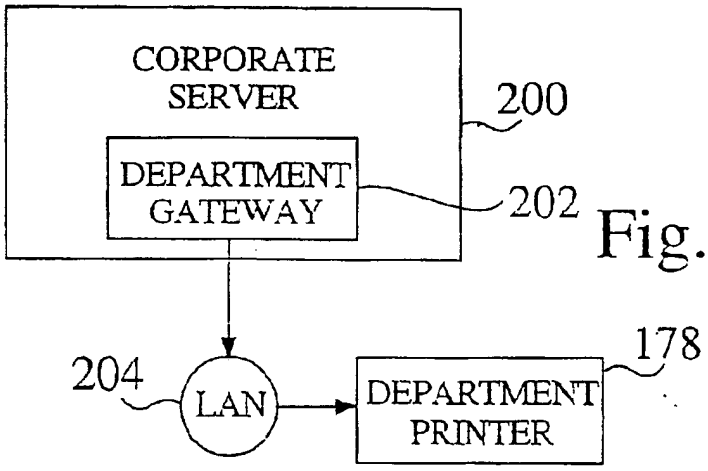


Fig. 21

13

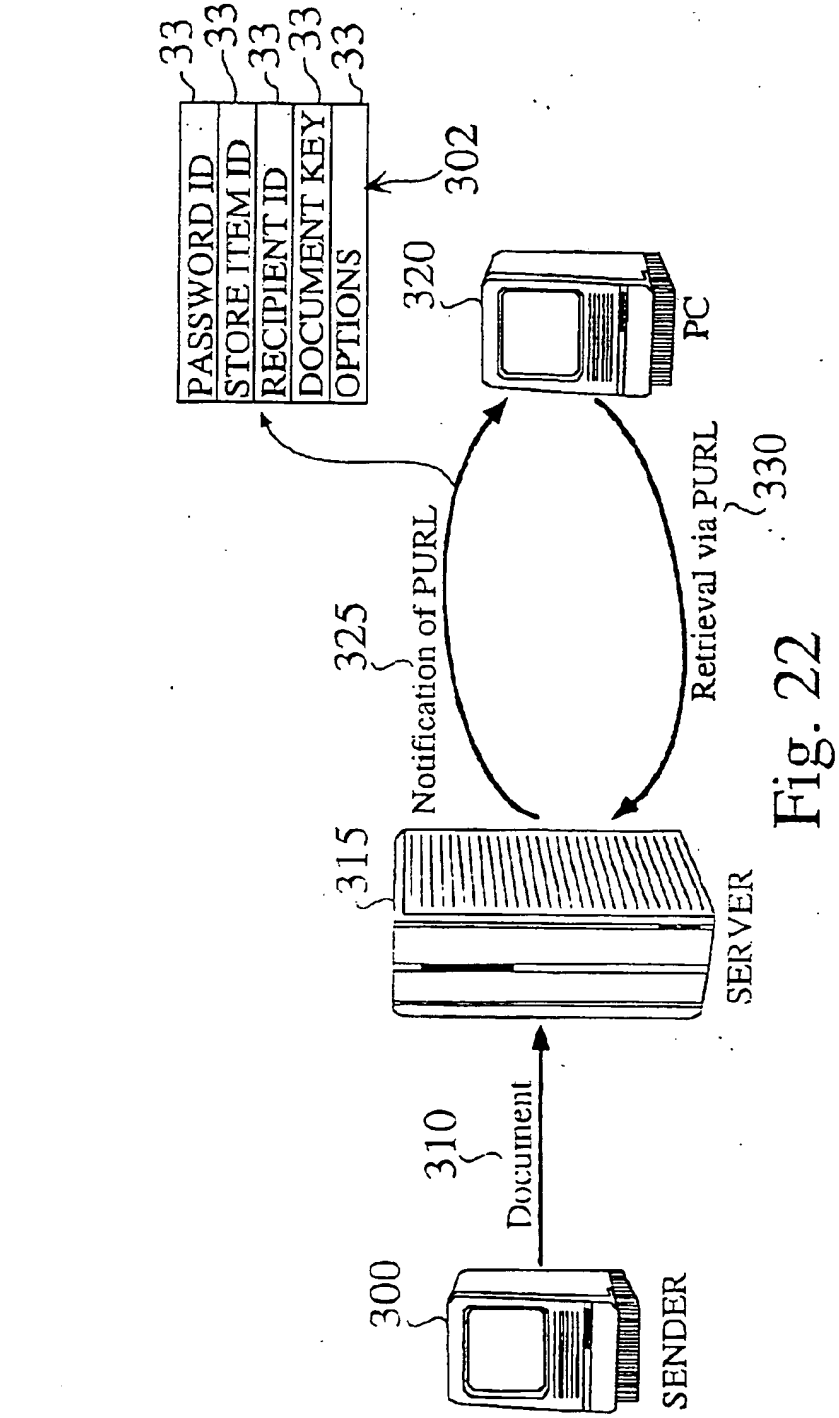

300
SENDER
310
Document
315
SERVER
325
Notification of PURL
330
Retrieval via PURL
320
PC
302
PASSWORD ID
STORE ITEM ID
RECIPIENT ID
DOCUMENT KEY
OPTIONS
33
33
33
33
33

Fig. 22

14

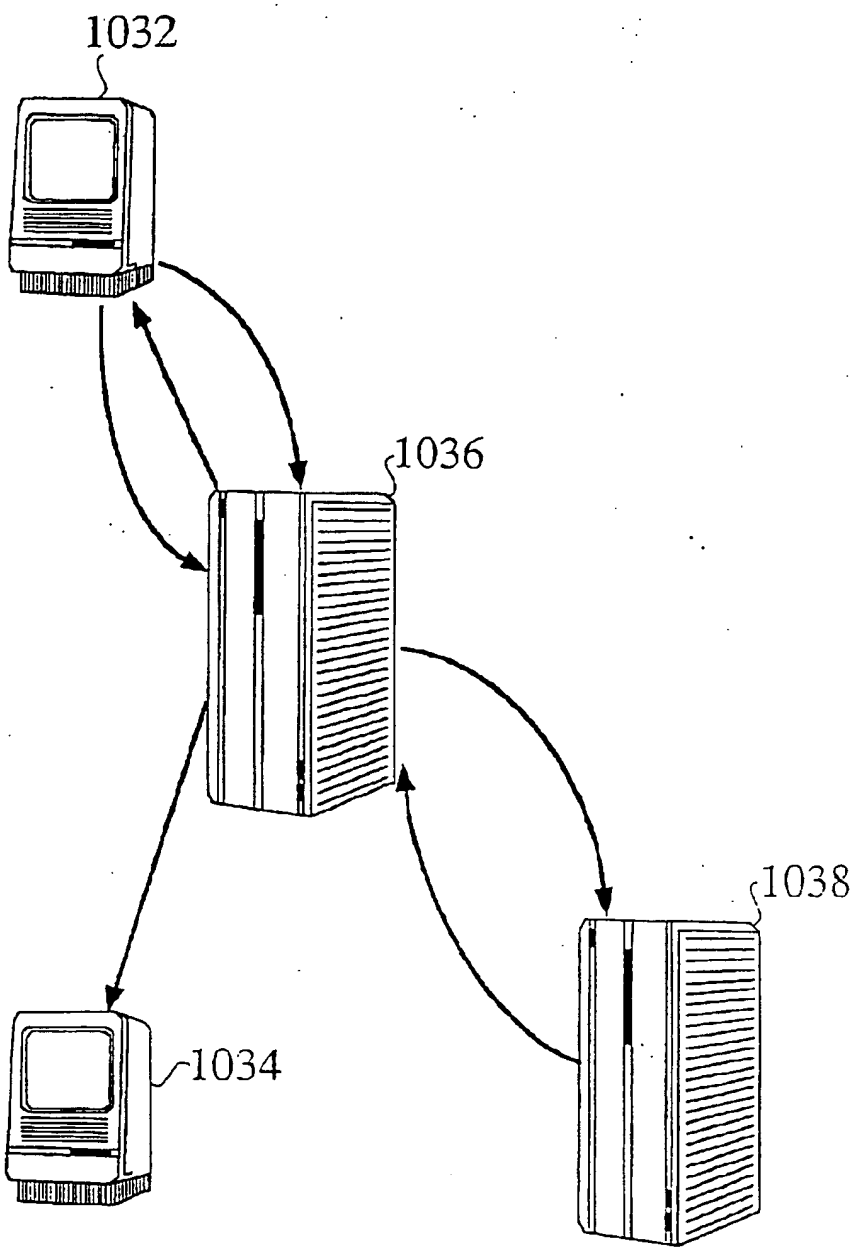


Fig. 23

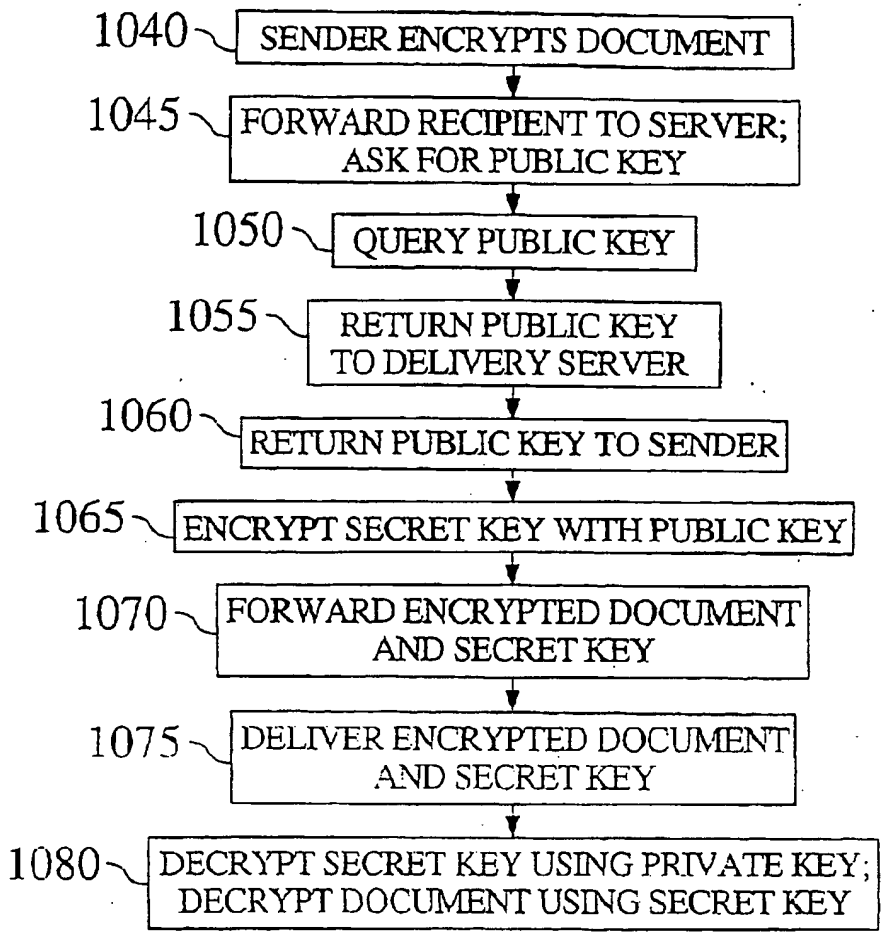

1040
SENDER ENCRYPTS DOCUMENT
1045
FORWARD RECIPIENT TO SERVER; ASK FOR PUBLIC KEY
1050
QUERY PUBLIC KEY
1055
RETURN PUBLIC KEY TO DELIVERY SERVER
1060
RETURN PUBLIC KEY TO SENDER
1065
ENCRYPT SECRET KEY WITH PUBLIC KEY
1070
FORWARD ENCRYPTED DOCUMENT AND SECRET KEY
1075
DELIVER ENCRYPTED DOCUMENT AND SECRET KEY
1080
DECRYPT SECRET KEY USING PRIVATE KEY; DECRYPT DOCUMENT USING SECRET KEY

Fig. 24

16

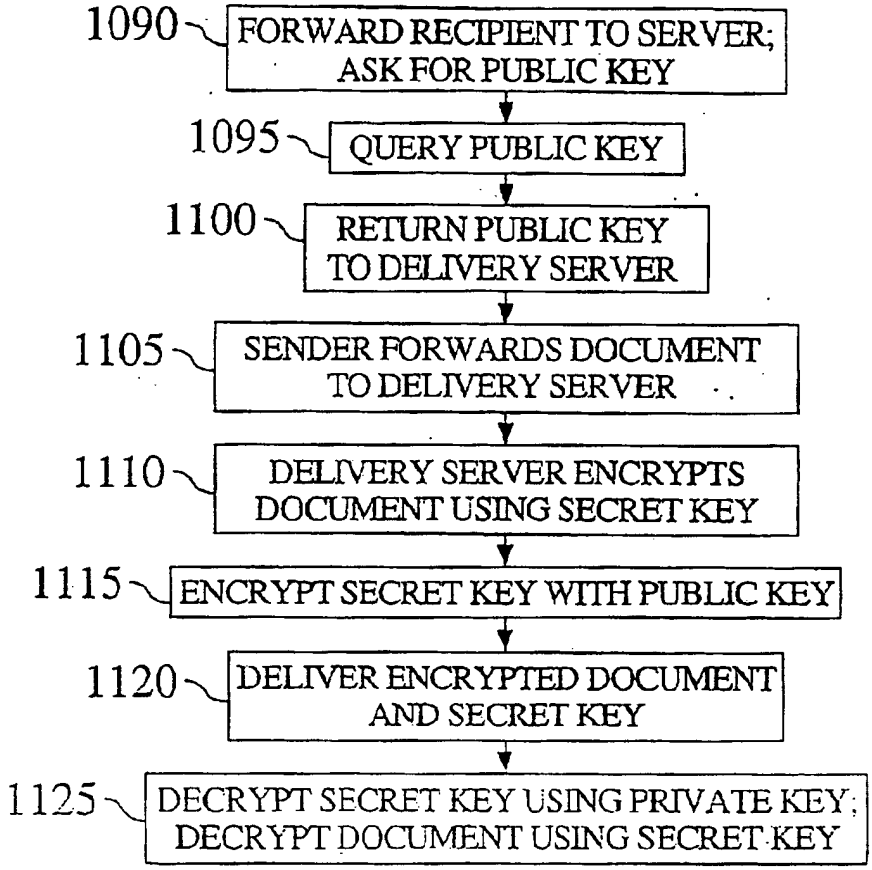

1090
FORWARD RECIPIENT TO SERVER; ASK FOR PUBLIC KEY
1095
QUERY PUBLIC KEY
1100
RETURN PUBLIC KEY TO DELIVERY SERVER
1105
SENDER FORWARDS DOCUMENT TO DELIVERY SERVER
1110
DELIVERY SERVER ENCRYPTS DOCUMENT USING SECRET KEY
1115
ENCRYPT SECRET KEY WITH PUBLIC KEY
1120
DELIVER ENCRYPTED DOCUMENT AND SECRET KEY
1125
DECRYPT SECRET KEY USING PRIVATE KEY; DECRYPT DOCUMENT USING SECRET KEY

Fig. 25

ABSTRACT OF THE DISCLOSURE

A document delivery architecture dynamically generates a private Uniform Resource Locator (URL) to distribute information. Each private URL ("PURL") uniquely identifies an intended recipient of a document, the document or set of documents to be delivered, and (optionally) other parameters specific to the delivery process. The intended recipient of a document uses the PURL to retrieve the document. The server, upon retrieval of the document, customizes the behavior of the retrieval based upon attributes included in the PURL, as well as log information associated with the retrieval in a data base. This architecture and usage of PURLs enables secure document delivery and tracking of document receipt. A method and system are provided for secure document delivery over a wide area network, such as the Internet. A sender directs a Delivery Server to retrieve an intended recipient's public key. The Delivery Server dynamically queries a certificate authority and retrieves the public key. The public key is transmitted from the Delivery Server to the sender. The sender encrypts the document using a secret key and then encrypts the secret key using the public key. Both encrypted document and encrypted secret key are uploaded to the Delivery Server, and transmitted to the intended recipient. The intended recipient then uses the private key associated with the public key to decrypt the secret key, and uses the secret key to decrypt the document. In an alternative, equally preferred embodiment of the invention, the sender uses the public key to encrypt the document. In yet another embodiment, the server transmits the document to the Delivery Server for encryption.